

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-114862

(P2010-114862A)

(43) 公開日 平成22年5月20日(2010.5.20)

(51) Int.Cl.
H03M 13/19 (2006.01)F I
H03M 13/19テーマコード (参考)
5J065

審査請求 未請求 請求項の数 3 O L (全 44 頁)

(21) 出願番号	特願2008-290022 (P2008-290022)	(71) 出願人	000005821
(22) 出願日	平成20年11月12日 (2008.11.12)		パナソニック株式会社
(31) 優先権主張番号	特願2008-264382 (P2008-264382)		大阪府門真市大字門真1006番地
(32) 優先日	平成20年10月10日 (2008.10.10)	(74) 代理人	100105050
(33) 優先権主張国	日本国 (JP)		弁理士 鷲田 公一
		(72) 発明者	村上 豊
			大阪府門真市大字門真1006番地 パナソニック株式会社内
		(72) 発明者	岡村 周太
			大阪府門真市大字門真1006番地 パナソニック株式会社内
		Fターム(参考)	5J065 AD01 AD07 AH01

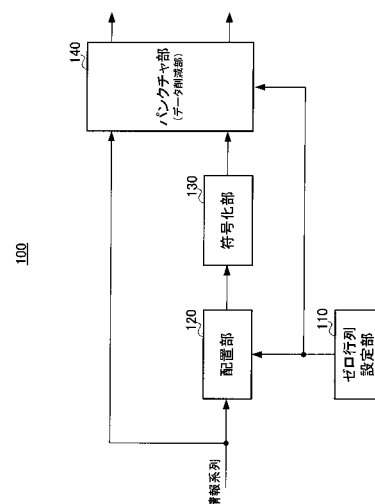
(54) 【発明の名称】 符号化器、送信装置及び符号化方法

(57) 【要約】

【課題】QC-LDPC等のブロック符号を用いる場合に、受信品質を向上させつつ、伝送量を低減させ、伝送効率の劣化を抑圧すること。

【解決手段】ゼロ行列設定部110は、パリティ生成行列gの部分行列であって、要素が全て“0”から構成されるゼロ行列を設定する。配置部120は、ゼロ行列の列に入力ビットを配置し、ゼロ行列以外の列に“0”を配置する。符号化部130は、パリティ生成行列gを用いて符号化してパリティビットを取得する。パンクチャ部(データ削減部)140は、得られたパリティビットのうち、ゼロ行列の行に対応するパリティビットを、送信しないビットとしてパンクチャする。

【選択図】図6



【特許請求の範囲】**【請求項 1】**

情報ビットを入力し、前記情報ビットと Q C - L D P C のパリティ生成行列との行列演算によりパリティビットを生成する符号化器であって、

前記情報ビットにゼロを挿入し、前記情報ビット及び前記ゼロと、前記パリティ生成行列との行列演算により前記パリティビットを生成し、

前記情報ビットを配置する位置と前記パリティ生成行列とに基づいて、前記パリティビットのうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力する、

符号化器。

10

【請求項 2】

前記請求項 1 に記載の符号化器を具備し、

前記パリティビットのうち、前記値が常にゼロとなるパリティビット以外の前記パリティビットと、前記情報ビットとを送信する送信手段と、

を具備する送信装置。

【請求項 3】

情報ビットを入力し、前記情報ビットと Q C - L D P C のパリティ生成行列との行列演算によりパリティビットを生成する符号化方法であって、

前記情報ビットにゼロを挿入し、前記情報ビット及び前記ゼロと、前記パリティ生成行列との行列演算によりパリティビットを生成し、

20

前記情報ビットを配置する位置と前記パリティ生成行列とに基づいて、前記パリティビットのうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力する、

符号化方法。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、例えば、Q C - L D P C (Quasi Cyclic Low Density Parity Check) 符号等のように、ゼロ行列を部分的かつ規則的に含むパリティ生成行列を用いて符号化系列を形成する符号化器、送信装置及び符号化方法に関する。

30

【背景技術】**【0002】**

近年、実現可能な回路規模で高い誤り訂正能力を発揮する誤り訂正符号として、低密度パリティ検査 (L D P C : Low Density Parity Check) 符号に注目が集まっている。L D P C 符号は、低密度なパリティ検査行列 H で定義される誤り訂正符号である。なお、低密度とは、行列中に含まれる 1 の要素数が 0 の要素数に比べて大幅に少ないことである。L D P C 符号は、検査行列 H の列数 N と等しいブロック長をもつブロック符号である。

【0003】

L D P C 符号は、その誤り訂正能力の高さと、実装の容易さことから、IEEE802.11n の高速無線 L A N (Local Area Network) システム、デジタル放送システムなどの誤り訂正符号化方式に採用されている。また、ホームネットワークでも、Q C (Quasi Cyclic: 擬似巡回) - L D P C 符号の採用が検討されている。

40

【0004】

ブロック符号では、ブロック符号長が長いほど、誤り訂正能力が向上するという特徴がある。例えば、ヘッダのように、制御情報等を送信するシンボルを確実に伝送したい場合には、ヘッダ長に比べ符号長が長いブロック符号を用いることで、ヘッダの受信品質を確保することができる。

【0005】

また、誤り訂正符号は、情報を伝送するために用いる誤り訂正符号とヘッダを伝送するために用いる誤り訂正符号とを共通にした場合、回路規模の点で有利となる。なお、本願

50

では、制御情報等を送信するシンボルを「ヘッダ」と呼び説明するが、制御情報等を送信するシンボルを、例えば、制御シンボル（制御チャネル又は制御信号）、プリアンブル、テイルシンボル、パイロットシンボル（パイロットチャネル又はパイロット信号）、トレーニングシンボル等と呼んでも良い。

【 0 0 0 6 】

このとき、図 1 に示すように、送信する必要がある情報ビット数（例えば、ヘッダ長）が、ブロック符号のブロック長に比べ短いような場合、ブロック長の残りの部分の情報ビットを“ 0 ”と仮定して、符号化を行い、パリティビットを生成する。

【 0 0 0 7 】

そして、実際に送信する符号化系列としては、例えば、図 1 に示すように、送信する必要がある情報ビット（例えば、ヘッダ）及びパリティビットのみを送信する。すなわち、“ 0 ”と仮定した情報ビットの部分は、実際には送信しないようにする。

【 0 0 0 8 】

一般に、制御情報等のヘッダは、画像等の情報を伝送するペイロードデータに比べビット数が少ない。しかし、図 1 に示すようにして、ヘッダ及びパリティビットを送信することによって、ヘッダとペイロードデータとを同一のブロック符号を用いて符号化することが可能となる。更に、ヘッダは、ヘッダ長より長いブロック長のブロック符号により符号化されるため、ヘッダの受信品質を確保することができる。この結果、ヘッダを通信相手に確実に伝送することができるようになるので、上述の通信方法は、通信を確立させるために有効である。

【非特許文献 1】「Rate-Compatible LDPC符号のレート推定法」、電子情報通信学会論文誌2006/12 Vol.J89 A NO.12 p.1177

【非特許文献 2】M. P. C. Fossorier, “Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” IEEE Trans. Inform. Theory, vol.50, no. 8, pp.1788-1793, Nov. 2001.

【非特許文献 3】L. Chen, J. Xu, I. Djurdjevic, and S. Lin, “Near-Shannon limit quasi-cyclic low-density parity-check codes,” IEEE Trans. Commun., vol.52, no.7, pp.1038-1042, July 2004.

【非特許文献 4】IEEE Unapproved Draft Std P802.11n_D3.00, pp.274, Sep 2007

【非特許文献 5】D. J. C. Mackay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inform. Theory, vol.45, no.2, pp399-431, March 1999.

【非特許文献 6】M. P. C. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low density parity check codes based on belief propagation,” IEEE Trans. Commun., vol.47, no.5, pp.673-680, May 1999.

【非特許文献 7】J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Yu Hu, “Reduced-complexity decoding of LDPC codes,” IEEE Trans. Commun., vol. 53., no.8, pp.1288-1299, Aug. 2005.

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 9 】

しかしながら、従来技術は、ヘッダのようにブロック長に比べデータ長が短いデータに対し、符号化を行うと、受信品質を向上することができるものの、情報ビットを“ 0 ”と仮定して符号化して得られたパリティビットをも送信しなくてはならない。そのため、ヘッダ長とブロック長とが同等で、その長さが短い場合には、送信しなければならないパリティビット数が少なく済む。これに対し、従来技術は、ヘッダ長よりブロック長が長い場合には、送信しなければならないパリティビット数が増え、データの伝送効率が劣化してしまうという課題がある。したがって、データ伝送効率が劣化する点を改善すると、データ伝送効率の向上と、受信品質の向上との両立を図ることができるという利点をもつことになる。

【 0 0 1 0 】

本発明はかかる点に鑑みてなされたものであり、例えば、QC-LDPC符号等のブロック符号を用いる場合に、受信品質を向上させつつ、伝送量を低減させ、伝送効率の劣化を抑圧することができる符号化器、送信装置及び符号化方法を提供することを目的とする。

【課題を解決するための手段】

【0011】

本発明の符号化器は、情報ビットを入力し、前記情報ビットとQC-LDPCのパリティ生成行列との行列演算によりパリティビットを生成する符号化器であって、前記情報ビットにゼロを挿入し、前記情報ビット及び前記ゼロと、前記パリティ生成行列との行列演算により前記パリティビットを生成し、前記情報ビットを配置する位置と前記パリティ生成行列とに基づいて、前記パリティビットのうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力する、構成を採る。

10

【0012】

本発明の送信装置は、上記符号化器を具備し、前記パリティビットのうち、前記値が常にゼロとなるパリティビット以外の前記パリティビットと、前記情報ビットとを送信する送信手段と、を具備する構成を採る。

【0013】

本発明の符号化方法は、情報ビットを入力し、前記情報ビットとQC-LDPCのパリティ生成行列との行列演算によりパリティビットを生成する符号化方法であって、前記情報ビットにゼロを挿入し、前記情報ビット及び前記ゼロと、前記パリティ生成行列との行列演算によりパリティビットを生成し、前記情報ビットを配置する位置と前記パリティ生成行列とに基づいて、前記パリティビットのうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力するようにした。

20

【発明の効果】

【0014】

本発明の通信装置及び通信方法によれば、例えば、QC-LDPC符号等のブロック符号を用いる場合に、受信品質を向上させつつ、伝送量を低減させ、伝送効率の劣化を抑圧することができる。

【発明を実施するための最良の形態】

【0015】

30

以下、本発明の実施の形態について、図面を参照して詳細に説明する。

【0016】

(実施の形態1)

図2は、本発明の通信装置に用いられる符号化器の入出力データを示す。図2の符号化器100は、QC-LDPC(Quasi Cyclic Low Density Parity Check)符号を形成する。

【0017】

図2において、情報系列 $u = (x_1, x_2, \dots, x_m)$ は、符号化器100の入力データであり、符号化系列 $s = (x_1, x_2, \dots, x_m, p_1, p_2, \dots, p_n)$ は、符号化器の出力データを示す。

40

【0018】

式(1)は、QC-LDPC符号のパリティ検査行列 H を示す(非特許文献1、非特許文献2、非特許文献3参照)。

【数1】

$$H = \begin{bmatrix} I(p_{0,0}) & I(p_{0,1}) & \cdots & I(p_{0,L-1}) \\ I(p_{1,0}) & I(p_{1,1}) & \cdots & I(p_{1,L-1}) \\ \vdots & \vdots & \ddots & \vdots \\ I(p_{J-1,0}) & I(p_{J-1,1}) & \cdots & I(p_{J-1,L-1}) \end{bmatrix} \quad \cdots (1)$$

50

【 0 0 1 9 】

式 (1) において、 $0 \leq j \leq J - 1$ 、 $0 \leq l \leq L - 1$ であり、符号長 $N = p \times L$ (p は自然数) のパリティ検査行列 H である。また、サブブロック行列 $I(p_j, l)$ は、 q 行 r 列 ($r = (q + p_j, l) \bmod p$ ($0 \leq q \leq p - 1$)) が 1 であり、その他は “ 0 ” であるような巡回置換行列である。なお、 p_j, l は、乱数により “ 0 ” 又は “ 1 ” に決定される。

【 0 0 2 0 】

図 2 の符号化器 100 は、生成行列 G を用いて、符号化系列を生成する。ここで、生成行列 G は、パリティ検査行列 H と、式 (2) の関係がある。

【 数 2 】

$$GH^T = 0 \quad \cdots (2)$$

【 0 0 2 1 】

符号化系列 s は、情報系列 u 及び生成行列 G より、 $s^T = Gu^T$ と示すことができる。 $QC-LDPC$ 符号は、組織符号であるので、生成行列 G は、式 (3) のように示すことができる。

【 数 3 】

$$G = \begin{bmatrix} I \\ g \end{bmatrix} \quad \cdots (3)$$

【 0 0 2 2 】

ここで、 I は、 $m \times m$ の単位行列である。また、行列 g は、符号化系列 s のうち、パリティ系列 w のみを取り出し、 $w = (p_1, p_2, \dots, p_n)$ と定義した場合に、パリティ系列 w を求めるための行列 (パリティ生成行列) である。パリティ生成系列 w は、 $w^T = gu^T$ を満たす。

【 0 0 2 3 】

図 3 は、 $QC-LDPC$ 符号のパリティ生成行列 g の一例を示す。図 3 に示す $QC-LDPC$ 符号は、非特許文献 4 の Table 20-14 (LDPC parameters) に記載の符号化率 (Coding rate(R)) = $1/2$ 、 $LDPC$ 符号情報ブロック長 (LDPC code information block length (bits)) = 648、 $LDPC$ 符号語ブロック長 (LDPC codeword block length (bits)) = 1296 の $QC-LDPC$ 符号である。

【 0 0 2 4 】

図 3 に示すパリティ生成行列 g は、複数のサブブロック行列 $201, 202, \dots, 211, 212, \dots$ から構成される。例えば、図 3 のサブブロック行列 201 において、 $(i + 1)$ 行目の各要素は、 i 行目の各要素を 1 ビット (1 列) 右にシフトした値をとる (i は自然数)。同様に、図 3 のサブブロック行列 211 において、 $(i + 1)$ 行目の各要素は、 i 行目の各要素を 1 ビット右にシフトした値をとる (i は自然数)。

【 0 0 2 5 】

また、図 3 のサブブロック行列 202 において、2 行目の各要素は、1 行目の各要素を 1 ビット右にシフトした値をとる。同様に、図 3 のサブブロック行列 212 において、2 行目の各要素は、1 行目の各要素を 1 ビット右にシフトした値をとる。

【 0 0 2 6 】

このように、サブブロック行列 $201, 202, \dots, 211, 212$ は、巡回置換行列といえる。図 3 に示す例では、サブブロック行列 $201, 202, \dots, 211, 212$ は、 27 行 27 列の行列である。

【 0 0 2 7 】

更に、パリティ生成行列 g において列が同一のサブブロック行列同士は、関連性を有している。例えば、サブブロック行列 201 と、サブブロック行列 201 と列が同一のサブブロック行列 211 とを比較すると、サブブロック行列 211 の i 行目の要素は、サブブ

ロック行列 2 0 1 の $(i + 1)$ 行目の要素と 2 ビット目が異なるだけである (i は自然数)。

【0028】

同様に、サブブロック行列 2 0 2 と、サブブロック行列 2 0 2 の列とが同一のサブブロック行列 2 1 2 とを比較すると、サブブロック行列 2 1 2 の i 行目の要素は、サブブロック行列 2 0 1 の $(i + 1)$ 行目の要素と同じである (i は自然数)。

【0029】

なお、27 行 27 列のサブブロック行列を縦に眺めた場合、例えば、サブブロック行列 2 0 1 とサブブロック行列 2 1 1 とを眺めた場合、上述したように、これらサブブロック行列は、関連性を有しているものの、必ずしも同一の行列であるとは限らない。

10

【0030】

更に、パリティ生成行列 g では、要素 “0” が連続して配置されるのが特徴的である。そのため、図 3 と同一のパリティ生成行列 g を示す図 4 から分かるように、サブブロック行列 2 0 2 内に、行列を構成する要素が全て “0” の行列 2 2 1 を確保することができる。以下、行列を構成する要素が全て “0” の行列を、ゼロ行列と呼ぶ。

【0031】

また、サブブロック行列 2 1 2 内には、ゼロ行列 2 2 1 と同じ列から始まり、ゼロ行列 2 2 1 と列数が同じ大きさのゼロ行列 2 2 2 を確保することができる。ゼロ行列 2 2 1 と同じ列から始まり、ゼロ行列 2 2 1 と列数が同じ大きさのゼロ行列は、図示せぬパリティ生成行列 g 内に多数存在する。

20

【0032】

このように、QC-LDPC 符号のパリティ生成行列は、ゼロ行列を含み、かつ、パリティ生成行列の同じ列から始まるゼロ行列が、多数存在するのが特徴的である。

【0033】

本発明者らは、QC-LDPC 符号のパリティ生成行列 g のこの特徴に着目した。すなわち、 m 行 n 列のゼロ行列以外の列に情報ビットとして “0” を配置すると、生成される m 個のパリティビットが、全てゼロになることに着目した。更に、パリティ生成行列 g において列が同一のサブブロック行列同士は、要素の配列に関連性があり、QC-LDPC 符号のパリティ生成行列 g には、同じ列から始まるゼロ行列が多数存在するため、ゼロ行列以外の列に情報ビットとして “0” を配置することにより、全てゼロになるパリティビットが多数生成されることに着目した。

30

【0034】

つまり、送信する必要がある情報ビット数が、ブロック符号のブロック長に比べ短く、一部の情報ビットを “0” と仮定して符号化する場合に、送信する必要がある情報ビットをゼロ行列 (m 行 n 列) の列に配置し、仮想ビットとして “0” をゼロ行列 (m 行 n 列) 以外に配置すると、値が “0” となる m 個のパリティビットが生成される。これらのパリティビットは、送信する必要がある情報ビットに依存せず、必ず “0” となる。

【0035】

したがって、受信側は、ゼロ行列の位置から、値が必ず “0” の m 個のパリティビットの位置が分かるので、送信側からは、値が必ず “0” の m 個のパリティビットが送信せずとも、受信側は全てのデータの復号することができる。また、受信側は、値が必ず “0” の m 個のパリティビット分を送信装置が送信しないビットと設定することができる、つまり、冗長ビットとして削減することができる。

40

【0036】

再度、図 4 を用いて詳細に説明する。図 4 のゼロ行列 2 2 1 に着目する。図 4 のゼロ行列 2 2 1 は、7 行 12 列の行列であり、ゼロ行列 2 2 1 の列に対応する情報ビットは、 $x_{36} \sim x_{47}$ である。そこで、 $x_{36} \sim x_{47}$ に、送信する必要がある情報ビットを配置し、 $x_{36} \sim x_{47}$ 以外に情報ビット “0” を配置して符号化を行うと、 $p_1 \sim p_7$ は、 $x_{36} \sim x_{47}$ の値に関わらず、必ず “0” となる。

【0037】

50

同様に、パリティ生成行列 g における列の位置が、ゼロ行列 2 2 1 と同じであるゼロ行列 2 2 2 に着目すると、 $x_{36} \sim x_{47}$ 以外に情報ビット “0” を配置して符号化を行うと、 $p_{28} \sim p_{34}$ は、 $x_{36} \sim x_{47}$ の値に関わらず、必ず “0” となる。

【0038】

したがって、パリティ生成行列 g により生成されたパリティビット $p_1 \sim p_{54}$ のうち、 $p_1 \sim p_7$ 及び $p_{28} \sim p_{34}$ は、値が必ず “0” となる。このため、送信装置が、パリティ生成行列 g により生成された、これら値が必ず “0” となるパリティビット $p_1 \sim p_{54}$ を送信しないようにすると、送信装置が送信する必要があるビットとしては、 $x_{36} \sim x_{47}$ 及び $p_8 \sim p_{27}$ 、 $p_{35} \sim p_{54}$ のみとすることができる。なお、以上の説明では、一例として、 $p_1 \sim p_{54}$ に着目して説明したが、 p_{55} 以降についても同様に考えることで、送信装置が送信するパリティビットの数の削減を行うことができる。

【0039】

図 4 のゼロ行列 2 2 1 は、7 行 1 2 列であるため、送信する必要がある情報ビットが 1 2 ビット以下の場合には、ゼロ行列 2 2 1 の列に送信する必要がある情報ビットを配置すれば良い。

【0040】

なお、送信する必要がある情報ビットが、1 2 ビットを超える場合には、例えば、図 5 に示すように、更にゼロ行列 2 3 1、2 3 2 ... の列に送信する必要がある情報ビットを配置するようにすれば良い。QC-LDPC 符号のパリティ生成行列の特徴として、“0” が連続して配置されているため、図 5 に示すように、パリティ生成行列 g には、ゼロ行列 2 2 1、2 2 2 ... 以外にも、ゼロ行列 2 3 1、2 3 2 のようなゼロ行列が多数存在する。

【0041】

ゼロ行列 2 3 1、2 3 2 は、7 行 7 列の行列であり、 $x_{71} \sim x_{77}$ に送信する必要がある情報ビットを配置した場合においても、 $p_1 \sim p_7$ 及び $p_{28} \sim p_{34}$ が全て “0” となる。そのため、送信装置は、ゼロ行列 2 2 1、2 2 2 を用いる場合と同様に、 $p_1 \sim p_7$ 及び $p_{28} \sim p_{34}$ を送信しなくても良い。

【0042】

したがって、ゼロ行列 2 2 1、2 2 2 に加え、ゼロ行列 2 3 1、2 3 2 を利用する場合には、送信する必要がある情報ビットを、 $x_{36} \sim x_{47}$ 及び $x_{71} \sim x_{77}$ に配置することができる。これにより、最大ビット数が、19 (= 12 + 7) ビットとなり、ゼロ行列 2 2 1、2 2 2 のみを利用する場合に比べ、送信する必要がある情報ビットとして配置可能な最大ビット数を増加することができる。

【0043】

同様に、送信する必要がある情報ビット数が、19 ビットを超える場合には、他の部分行列に含まれるゼロ行列を利用すれば良い。図 5 には、QC-LDPC 符号のパリティ生成行列 g の一部だけを示しており、QC-LDPC のパリティ生成行列 g には、2 7 行 2 7 列の巡回置換行列が列方向に 24 (= 648 / 27) 個存在するため、図示せぬ領域にもゼロ行列が多数含まれている。このため、送信装置は、ゼロ行列の部分において、上記と同様にゼロ行列を利用して、送信する必要がある情報ビットとして、配置することができる最大ビット数を増やすことができる。

【0044】

このように、QC-LDPC 符号のパリティ生成行列 g には、パリティ生成行列 g の同じ列から始まり、列数が同一のゼロ行列が、複数存在することに着目し、本実施の形態では、ゼロ行列の列に送信する必要がある情報ビットを配置し、ゼロ行列以外の列に仮想ビットとして “0” を配置する。これにより、ゼロ行列の行数と同じ数だけ値が “0” となるパリティビットが生成されるようになる。

【0045】

このとき、送信装置及び受信装置は、パリティ生成行列 g に対するゼロ行列の位置を共有していれば、このゼロ行列の行に対応するパリティビットを実際に送信せずとも、受信側では “0” が送信されたとして復号処理を行うことにより、パリティ生成行列 g により

10

20

30

40

50

符号化された符号化データを復号することができる。このため、送信装置は、送信する必要があるパリティビット数を低減し、伝送効率を向上させることができる。

【0046】

なお、ゼロ行列は、1行1列であっても良い。すなわち、同一行に1行1列のゼロ行列が複数あり、この複数のゼロ行列と同一列の要素がゼロの行があれば、同一列において要素がゼロとなる行数分だけ、常に“0”となるパリティビットが生成されることになる。

【0047】

すなわち、送信装置は、情報ビットにゼロを挿入し、情報ビット及びゼロと、QC-LDPC符号のパリティ生成行列との行列演算によりパリティビットを生成する場合において、情報ビットを配置する位置とパリティ生成行列とに基づいて、パリティビットのうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力し、削除後のパリティ系列を送信するようにすることで、送信する必要があるパリティビット数を低減し、伝送効率を向上させることができる。

【0048】

なお、送信装置は、パリティ生成行列gの同じ列から始まり、列数が同一のゼロ行列（1行1列のゼロ行列も含む）のうち、行数が最も多い行列を、設定するゼロ行列とし、設定したゼロ行列の列以外に“0”を配置することにより、値が“0”となるパリティビットを、設定したゼロ行列の行数分だけ生成するようになる。

【0049】

したがって、送信装置は、値が“0”となるパリティビットを送信しないビットとしてパンクチャすることにより、伝送効率を向上させることができる。このとき、送信装置は、パリティ生成行列gの同じ列から始まり、列数が同一のゼロ行列のうち、パリティ生成行列gにより多く含まれるような部分行列をゼロ行列として設定することにより、パリティビットをより削減することができる。

【0050】

なお、このとき、送信装置は、送信する必要がある情報ビットとして、配置することができる最大ビット数を、ゼロ行列の列数とする。例えば、送信装置は、ゼロ行列として、ゼロ行列221, 222, ...が設定された場合、送信する必要がある情報ビットを配置することができる最大ビット数は12ビットとなる。

【0051】

また、送信装置は、ゼロ行列として、ゼロ行列221, 222に加え、ゼロ行列231, 232が設定される場合には、送信する必要がある情報ビットを配置することができる最大ビット数は19ビットとなる。逆に言えば、送信装置は、送信する必要がある情報ビットのデータ長（ビット数）に応じて、ゼロ行列を設定するようすれば良い。なお、上述したように、ゼロ行列は、1行1列でも良く、また、連続していなくても良い。

【0052】

図6には、上述のようなパリティ生成行列gを用いて符号化を行う符号化器の構成例を示す。図6の符号化器100は、ゼロ行列設定部110、配置部120、符号化部130及びパンクチャ部（データ削減部）140を有する。以下では、ヘッダ等のように、データ長が一定の情報系列が、符号化器100に入力される場合を例に説明する。

【0053】

ゼロ行列設定部110は、QC-LDPCのパリティ生成行列gの部分行列であって、要素が全て“0”から構成されるゼロ行列を設定する。ゼロ行列の設定方法は、情報系列のデータ長がヘッダ等のように一意に決まっている場合、列数がヘッダ長以上のゼロ行列を設定する。なお、以下では、図4のゼロ行列221, 222, ...が、ゼロ行列として設定された場合を例に説明する。ゼロ行列設定部110は、パリティ生成行列gにおけるゼロ行列の位置の情報を、配置部120及びパンクチャ部（データ削減部）140に出力する。

【0054】

配置部120は、ヘッダ等の情報系列を入力し、ゼロ行列設定部110から通知される

10

20

30

40

50

ゼロ行列の位置の情報に基づいて、ゼロ行列の列に情報ビット（入力ビット）を配置し、ゼロ行列以外の列に仮想ビットとして“0”を配置する。

【0055】

例えば、ゼロ行列設定部110からゼロ行列221, 222の位置が通知された場合、配置部120は、ゼロ行列221の列 $x_{36} \sim x_{47}$ に、情報ビット（入力ビット）を配置し、 $x_{36} \sim x_{47}$ 以外に、“0”を配置する。配置部120は、配置後のビットを符号化部130に出力する。

【0056】

符号化部130は、パリティ生成行列 g を用いて、配置部120から出力されるビットを符号化して符号化系列（情報ビット及びパリティビット）を取得する。符号化部130は、符号化系列をパンクチャ部（データ削減部）140に出力する。

10

【0057】

パンクチャ部（データ削減部）140は、ゼロ行列設定部110から通知されるゼロ行列221, 222の位置の情報に基づいて、符号化系列から、 $x_{36} \sim x_{47}$ 以外に配置された“0”を送信しないビットとしてパンクチャ（削除）する。

【0058】

また、パンクチャ部（データ削減部）140は、ゼロ行列設定部110から通知されるゼロ行列221, 222の位置の情報に基づいて、符号化系列から、ゼロ行列221, 222の行に対応するパリティビット $p_1 \sim p_7$, $p_{28} \sim p_{34}$, ...を送信しないビットとしてパンクチャ（削除）する。

20

【0059】

パンクチャ部（データ削減部）140は、符号化系列から送信しないビットとしてパンクチャ（削除）したビット以外の符号化系列を、送信する必要があるビットとして出力する。

【0060】

図7は、上述した符号化器から送信された信号を復号する復号化器の構成例を示す。

【0061】

復号化器300は、固定対数尤度比挿入部310及びBP（Belief Propagation）復号部320を有している。

【0062】

30

固定対数尤度比挿入部310は、図示せぬ対数尤度比算出部により算出される受信対数尤度比、及び、ゼロ行列の位置に関する情報を示す制御信号を入力し、ゼロ行列の位置に応じて、受信対数尤度比に既知の対数尤度比を挿入する。

【0063】

例えば、符号化側で、ゼロ行列221, 222, ...が用いられた場合、固定対数尤度比挿入部310には、 $x_{36} \sim x_{47}$ 及び $p_8 \sim p_{27}$, $p_{35} \sim \dots$ に対応する受信対数尤度比 $LLR_{x_{36}} \sim LLR_{x_{47}}$, $LLR_{p_8} \sim LLR_{p_{27}}$, $LLR_{p_{35}} \sim \dots$ が入力される。そこで、固定対数尤度比挿入部310は、 $x_1 \sim x_{35}$, x_{48} , ...に対応する受信対数尤度比 $LLR_{x_1} \sim LLR_{x_{35}}$, $LLR_{x_{48}} \dots$, $LLR_{p_1} \sim LLR_{p_7}$, $LLR_{p_{28}} \sim LLR_{p_{34}}$ を挿入する。

40

【0064】

具体的には、符号化側で、ゼロ行列221, 222, ...が用いられた場合、 $x_1 \sim x_{35}$, x_{48} , ..., $p_1 \sim p_7$, $p_{28} \sim p_{34}$, ...として“0”が送信されていることに相当するので、固定対数尤度比挿入部310は、既知ビット“0”に対応する固定の対数尤度比を、 $x_1 \sim x_{35}$, $x_{48} \dots$ の対数尤度比 $LLR_{x_1} \sim LLR_{x_{35}}$, $LLR_{x_{48}} \dots$, $LLR_{p_1} \sim LLR_{p_7}$, $LLR_{p_{28}} \sim LLR_{p_{34}} \dots$ として挿入する。図7において、点線の丸で囲まれた受信対数尤度比は、固定対数尤度比挿入部310によって挿入された受信対数尤度比を示す。

【0065】

固定対数尤度比挿入部310は、挿入後の対数尤度比をBP復号部320に出力する。

50

【 0 0 6 6 】

B P 復号部 3 2 0 は、例えば、非特許文献 5 から非特許文献 7 に記載された sum-product 復号、min-sum 復号、Normalized BP 復号、offset BP 復号などを用いて、復号する。

【 0 0 6 7 】

以下には、上述のように構成された符号化器を有する通信装置 # 1 の構成、及び、上述のように構成された復号化器を有し、通信装置 # 1 から送信された信号を受信する通信装置 # 2 の構成について説明する。

【 0 0 6 8 】

図 8 には、通信装置 # 1 が送信する変調信号のフレーム構成例を示す。制御情報シンボルは、通信相手（通信装置 # 2）に、変調方式、使用している誤り訂正符号、符号化率、送信方法、データ長等の制御情報を伝送するためのシンボルである。情報シンボルは、Q C - L D P C 符号化により得られた情報ビット及びパリティビットを伝送するためのシンボルである。

10

【 0 0 6 9 】

図 9 には、通信装置 # 1 の構成例を示す。図 9 の通信装置 4 0 0 において、符号化部 4 1 0 は、情報系列を入力し、符号化系列をインタリーバ 4 2 0 に出力する。符号化部 4 1 0 は、図 6 の符号化器 1 0 0 によって構成されている。

【 0 0 7 0 】

インタリーバ 4 2 0 は、符号化系列を入力し、インタリーブを行うことでインタリーブ後のデータを得る。なお、符号の種類によっては、インタリーバ 4 2 0 を設けなくても良い。

20

【 0 0 7 1 】

マッピング部 4 3 0 は、インタリーブ後のデータを入力し、Q P S K（Quadrature Phase Shift Keying）、1 6 Q A M（Quadrature Amplitude Modulation）等の変調を行うことでベースバンド信号を得る。

【 0 0 7 2 】

送信部 4 4 0 は、ベースバンド信号を入力し、直交変調、周波数変換等の所定の信号処理を施すことで変調信号を得、変調信号を送信する。

【 0 0 7 3 】

図 1 0 は、通信装置 # 2 の構成例を示す。図 1 0 の通信装置 5 0 0 において、受信部 5 1 0 は、受信信号を入力し、周波数変換等の所定の無線処理を施すことでベースバンド信号を得る。受信部 5 1 0 は、ベースバンド信号を、制御情報検出部 5 2 0 及び対数尤度比算出部 5 3 0 に出力する。

30

【 0 0 7 4 】

制御情報検出部 5 2 0 は、ベースバンド信号からゼロ行列に関する情報、インタリーブパターンの情報、符号化率に関する情報等を検出する。そして、制御情報検出部 5 2 0 は、インタリーブパターンの情報をデインタリーバ 5 4 0 に出力し、ゼロ行列に関する情報及び符号化率に関する情報を復号化部 5 5 0 に出力する。

【 0 0 7 5 】

対数尤度比算出部 5 3 0 は、ベースバンド信号を入力し、例えば非特許文献 5 に示されている方法を用いることで対数尤度比を算出し、ビットごとの対数尤度比を得る。対数尤度比算出部 5 3 0 は、ビットごとの対数尤度比をデインタリーバ 5 4 0 に出力する。

40

【 0 0 7 6 】

デインタリーバ 5 4 0 は、ビットごとの対数尤度比を入力し、インタリーバ 4 2 0 に対応するデインタリーブの処理を施すことでデインタリーブ後の対数尤度比を得る。なお、復号化部 5 5 0 は、B P 復号を行う場合に、デインタリーバ 5 4 0 を設けなくても、デインタリーブを考慮した検査行列を用意することで、復号することが可能である。

【 0 0 7 7 】

復号化部 5 5 0 は、図 7 の復号化器 3 0 0 によって構成されている。復号化部 5 5 0 は、デインタリーブ後の対数尤度比を入力し、符号化部 4 1 0 に対応する復号を行うことで

50

受信データを得る。

【 0 0 7 8 】

以上のように、本実施の形態では、ゼロ行列設定部 1 1 0 は、パリティ生成行列 g の部分行列であって、要素が全て “ 0 ” のゼロ行列を設定する。配置部 1 2 0 は、ゼロ行列の列に入力ビットを配置し、ゼロ行列以外の列に “ 0 ” を配置する。符号化部 1 3 0 は、パリティ生成行列 g を用いて符号化してパリティビットを取得する。パンクチャ部（データ削減部） 1 4 0 は、ゼロ行列設定部 1 1 0 から通知されるゼロ行列の位置の情報に基づいて、ゼロ行列以外の列に配置された “ 0 ” を送信しないビットとしてパンクチャ（削除）し、更に、得られたパリティビットのうち、ゼロ行列の行に対応するパリティビットを、送信しないビットとしてパンクチャ（削除）する。

10

【 0 0 7 9 】

これにより、符号化器 1 0 0 は、情報ビットを入力し、情報ビットとパリティ生成行列との行列演算によりパリティビットを生成する場合において、パリティ生成行列のうち、要素が全てゼロの部分行列の列に対応する位置に情報ビットを配置し、要素が全てゼロの部分行列以外の列に対応する位置にゼロを配置し、配置後の情報ビット及びゼロと、パリティ生成行列との行列演算をする。これにより、符号化器 1 0 0 は、パリティ系列を生成し、パリティ系列のうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力する。

【 0 0 8 0 】

換言すると、符号化器 1 0 0 は、情報ビットにゼロを挿入し、情報ビット及びゼロと、 $QC-LDPC$ のパリティ生成行列との行列演算によりパリティビットを生成し、ゼロを挿入する位置及びパリティ生成行列とに基づいて、パリティビットのうち、値が常にゼロとなるパリティビットを削除し、削除後のパリティ系列を出力する。

20

【 0 0 8 1 】

したがって、符号化器 1 0 0 を有する送信装置 4 0 0 において、送信部 4 4 0 が、入力ビットと、パリティビットのうち、ゼロ行列の行に対応する前記パリティビット以外と、を送信することにより、ゼロ行列の行に対応するパリティビットを受信側に送信せずとも、受信側では、ゼロ行列の行に対応するパリティビットの対数尤度比として既知の固定の対数尤度比を挿入して、復号を行うことができるため、送信するパリティビット数を低減し、伝送効率を向上させることができる。

30

【 0 0 8 2 】

なお、送信する必要がある情報ビットは、制御情報等を含むヘッダに限られず、ペイロードデータ（情報伝送用のシンボル）等であっても良い。要するに、送信する必要がある情報ビットの数が、 $QC-LDPC$ 符号に含まれるゼロ行列の列数に比べ小さければ、本発明を適用することができる。送信する必要がある情報ビットがヘッダであり、ヘッダ長が固定の場合には、ゼロ行列設定部 1 1 0 は、ヘッダ長に応じて予め最適なゼロ行列を設定しておくことができる。

【 0 0 8 3 】

一方、送信する必要がある情報ビットが、ペイロードデータの場合には、コンテンツ情報等の大きさによりデータ長が変動する。本発明は、送信する必要がある情報ビットのデータ長が、例えば、ペイロードデータのように変動する場合においても適用することができる。以下の実施の形態 2 では、送信する必要がある情報ビットのデータ長が変動する場合について説明する。

40

【 0 0 8 4 】

（実施の形態 2）

本実施の形態では、送信する必要がある情報ビットのデータ長が変動する場合に、本発明を適用した場合の形態について説明する。

【 0 0 8 5 】

図 1 1 は、 $QC-LDPC$ 符号を用いた場合の 1 ブロックの構成例を示す。 $QC-LDPC$ 符号は、ブロック符号であり、図 1 1 に示すように、1 ブロックは、情報ビットとパ

50

リティビットとから構成される。ここで、1ブロック内の情報ビットのビット数がMビットであるとする。

【0086】

図12は、本実施の形態に係る符号化器の構成例を示す。なお、図12の本実施の形態に係る符号化器において、図6と共通する構成部分には、図6と同一の符号を付して説明を省略する。図12の符号化器100aは、図6の符号化器100に対し、ゼロ行列設定部110及び配置部120に代えて、ゼロ行列設定部110a及び配置部120aを有している。以下では、Nビットの情報系列が符号化器100aに入力される場合を例に説明する。

【0087】

ゼロ行列設定部110aは、情報系列として入力される情報ビット（入力ビット）のデータ長Nに応じて、ゼロ行列を設定する。具体的には、ゼロ行列設定部110aは、まず、情報ビット（入力ビット）のデータ長Nをカウントする。そして、ゼロ行列設定部110aは、データ長NをQC-LDPC符号の1ブロックあたりの情報ビット長Mで除算し、商及び余りを算出する。

【0088】

配置部120aは、除算の結果、 $N = kM$ （kは整数）が成立する場合、図13に示すように、k個の全てのブロックの情報ビットの領域に、情報系列として入力される情報ビット（入力ビット）を配置する必要がある。すなわち、k個のブロックでは、QC-LDPC符号のパリティ生成行列gの全ての列に各情報ビット（入力ビット）を配置する必要がある。そのため、 $N = kM$ （kは自然数）が成立する場合は、ゼロ行列設定部110aは、ゼロ行列を設定せず、パリティ生成行列gの全ての列に情報ビット（入力ビット）を配置するよう配置部120aに指示信号を出力する。

【0089】

配置部120aは、除算の結果、 $N = kM + r$ （kは整数、r及びMは自然数）が成立する場合、図13に示すように、k個のブロックの情報ビットの領域には、情報ビット（入力ビット）を配置し、1個のブロック（特殊ブロック）の情報ビットの領域には、rビットの情報ビット（入力ビット）を配置する必要がある。すなわち、配置部120aは、k個のブロックでは、QC-LDPC符号のパリティ生成行列gの全ての列に各情報ビットを配置し、特殊ブロックでは、実施の形態1で述べたように、ゼロ行列の列に情報ビット（入力ビット）を配置するようになる必要がある。

【0090】

そのため、 $N = kM + r$ （k、r及びMは自然数）が成立する場合には、ゼロ行列設定部110aは、特殊ブロックで送信する必要がある情報ビット（入力ビット）のデータ長rに応じて、ゼロ行列を設定する。このとき、ゼロ行列設定部110aは、データ長rの値に応じて、設定するゼロ行列を切り替える。具体的には、ゼロ行列設定部110aは、余りrと所定の閾値との比較結果に応じて、設定するゼロ行列を切り替える。上述したように、符号化器100aは、ゼロ行列によって、送信する必要がある情報ビットとして配置できるビット数の最大値が変動する。

【0091】

なお、図13では、特殊ブロックを時間的に最後に配置しているが、配置位置はこれに限ったものではない。

【0092】

以下、図14を用いて、データ長に応じてゼロ行列を設定する動作について説明する。図14は、ゼロ行列設定部110aが、2つの閾値をa1、a2を有し、データ長rと2つの閾値との比較結果に応じて、ゼロ行列を切り替える場合の例である。送信しないビットとしてパンクチャできる（削減できる）パリティビット数は、ゼロ行列の行数と同じであるため、ゼロ行列を切り替えることは、換言すると、送信するパリティビットの削減方法を切り替えることになる。

【0093】

10

20

30

40

50

0 < a 1 の場合、ゼロ行列 # 1 (削減方法 # 1) により、送信するパリティビットを削減する。例えば、0 < a 1 (= 1 2) の場合、ゼロ行列設定部 1 1 0 a は、ゼロ行列 2 2 1, 2 2 2 ... をゼロ行列として設定する。

【0 0 9 4】

そして、= 1 0 の場合、配置部 1 2 0 a は、1 0 ビットの情報に“0”を2ビット加え、1 2 ビットにする。そして、配置部 1 2 0 a は、この1 2 ビットを、x 3 6 ~ x 4 7 に割り当て、x 1 ~ x 3 5, x 4 8 ~ に“0”を割り当てる。この結果、符号化部 1 3 0 により得られるパリティビットのうち、パリティビット p 1 ~ p 7, p 2 8 ~ p 3 4, ... は、x 3 6 ~ x 4 7 の値によらず常に“0”となる。

【0 0 9 5】

したがって、パンクチャ部 (データ削減部) 1 4 0 が、常に“0”となるパリティビット p 1 ~ p 7, p 2 8 ~ p 3 4 を送信しないビットとしてパンクチャすることにより、復号特性を劣化させることなく、伝送効率を向上させることができる。

【0 0 9 6】

また、x 3 6 ~ x 4 7 以外のビット (x 1 ~ x 3 5, x 4 8 ~) には、既知ビット“0”が割り当てられているので、パンクチャ部 (データ削減部) 1 4 0 は、これら x 3 6 ~ x 4 7 以外のビットもパンクチャする (送信しないビットと設定する)。加えて、= 1 0 の場合には、パンクチャ部 (データ削減部) 1 4 0 が、x 3 6 ~ x 4 7 に割り当てられた2ビットの“0”を送信しないビットとしてパンクチャする (送信しないビットと設定する)。これにより、伝送効率を更に向上させることができる。

【0 0 9 7】

例えば、配置部 1 2 0 a が、x 4 6, x 4 7 に“0”を割り当てた場合に、パンクチャ部 (データ削減部) 1 4 0 が、x 4 6, x 4 7 をパンクチャすることにより、送信系列は、x 3 6 ~ x 4 5、パリティ p 8 ~ p 2 7, p 3 5 ~ p 5 4, ... となり、伝送効率を更に向上させることができる。

【0 0 9 8】

a 1 < a 2 の場合、ゼロ行列 # 2 (削減方法 # 2) により、送信するパリティビットを削減する。例えば、a 1 = 1 2, a 2 = 1 9 の場合、ゼロ行列設定部 1 1 0 a は、ゼロ行列 2 2 1, 2 2 2, ... に加え、ゼロ行列 2 3 1, 2 3 2, ... を、ゼロ行列として設定する。

【0 0 9 9】

そして、= 1 5 の場合、配置部 1 2 0 a は、1 5 ビットの情報に“0”を4ビット加え、1 9 ビットにする。そして、配置部 1 2 0 a は、この1 9 ビットを、x 3 6 ~ x 4 7 及び x 7 1 ~ x 7 7 に割り当て、x 1 ~ x 3 5, x 4 8 ~ x 7 1, x 7 8 ~ に“0”を割り当てる。この結果、符号化部 1 3 0 により得られるパリティビットのうち、パリティビット p 1 ~ p 7, p 2 8 ~ p 3 4, ... は、x 3 6 ~ x 4 7 の値によらず常に“0”となる。

【0 1 0 0】

したがって、パンクチャ部 (データ削減部) 1 4 0 は、常に“0”となるパリティビット p 1 ~ p 7, p 2 8 ~ p 3 4 を送信しないビットとしてパンクチャすることにより、復号特性を劣化させることなく、伝送効率を向上させることができる。

【0 1 0 1】

また、x 3 6 ~ x 4 7, x 7 1 ~ x 7 7 以外のビット (x 1 ~ x 3 5, x 4 8 ~ x 7 1, x 7 8 ~) には既知ビット“0”が割り当てられているので、パンクチャ部 (データ削減部) 1 4 0 は、これら x 3 6 ~ x 4 7, x 7 1 ~ x 7 7 以外のビットをパンクチャする (送信しないビット) と設定する。加えて、= 1 5 の場合には、パンクチャ部 (データ削減部) 1 4 0 が、x 3 6 ~ x 4 7, x 7 1 ~ x 7 7 のいずれかに割り当てられた4ビットの“0”を送信しないビットとしてパンクチャする (送信しないビットと設定する)。

【0 1 0 2】

これにより、送信装置は、伝送効率を更に向上させることができる。例えば、配置部 1

10

20

30

40

50

20 a が、 $x_{74} \sim x_{77}$ に“0”を割り当てた場合に、パンクチャ部（データ削減部）140が、 $x_{74} \sim x_{77}$ をパンクチャすることにより、送信系列は、 $x_{36} \sim x_{45}$ 、 $x_{71} \sim x_{73}$ 、 $p_8 \sim p_{27}$ 、 $p_{35} \sim p_{54}$ 、...となり、送信装置は、伝送効率を更に向上させることができる。

【0103】

なお、図14に示す例では、 $a_2 < M - 1$ の場合、ゼロ行列を設定せず、パリティビットの削減は行わないようにした。すなわち、情報ビット（入力ビット）のデータ長Nをブロック長Mで除算した余りが所定の閾値以上の場合には、個の情報ビット（入力ビット）、及び、仮想ビットとして（ $M -$ ）個の“0”を、パリティ生成行列gの列に配置するようにする。

10

【0104】

このようにして、ゼロ行列設定部110aは、特殊ブロックで送信する必要がある情報ビット（入力ビット）のデータ長に応じて、ゼロ行列を設定する。そして、ゼロ行列設定部110aは、パリティ生成行列gにおけるゼロ行列の位置の情報を配置部120a及びパンクチャ部（データ削減部）140に通知する。

【0105】

なお、ゼロ行列設定部110aは、 $a_2 < M - 1$ の場合、ゼロ行列を設定せず、パリティビットの削減は行わないようにした。そのため、そのため、 $a_2 < M - 1$ の場合、ゼロ行列設定部110aは、パリティビットをパンクチャしないようパンクチャ部（データ削減部）140に通知する。

20

【0106】

以上のように、本実施の形態では、ゼロ行列設定部110aは、パリティ生成行列gの部分行列であって、要素が全て“0”のゼロ行列を、情報ビット（入力ビット）のデータ長Nに応じて設定するようにした。このようにすることで、送信装置は、送信する必要があるパリティビット数を低減しつつ、情報ビット（入力ビット）を確実に送信することができる。

【0107】

なお、図14には、余りの値に応じて、3つのいずれかに場合分けをする例を示したが、場合分け数は3つに限られない。例えば、ゼロ行列設定部110aが、閾値を更に有し、2個に場合分けするようにしても良い。

30

【0108】

また、本実施の形態を実現するにあたり、復号化器を具備する受信装置が、余りの値を知っている必要がある。これを実現する簡単な方法としては、符号化器を具備する送信装置が、送信するデータのビット数の情報を、最初に受信装置に通知すれば良い。なおこのとき、受信装置は、を求めるための演算部を具備する必要がある。

【0109】

（実施の形態3）

本実施の形態では、QC-LDPC符号におけるパンクチャ方法について説明する。

【0110】

図15は、本実施の形態に係る符号化器の構成例を示す。図15の符号化器600は、符号化部610、パンクチャパターン設定部620及びパンクチャ部（データ削減部）630を有している。

40

【0111】

符号化部610は、QC-LDPCのパリティ生成行列gを用いて、情報系列に符号化を行う。

【0112】

パンクチャパターン設定部620は、QC-LDPC符号の検査行列Hが、サブブロック行列を基本単位として構成されていることを利用して、パンクチャパターンを探索し、設定する。パンクチャパターンの探索方法については、後述する。パンクチャパターン設定部620は、設定したパンクチャパターンの情報を、パンクチャ部（データ削減部）6

50

30に出力する。

【0113】

バンクチャ部（データ削減部）630は、バンクチャパターン設定部620から通知されるバンクチャパターンに従って、符号化部610から出力される符号化系列のうち、送信しないビットとして情報ビット又はパリティビットをバンクチャする（送信しないビットと設定する）。

【0114】

次に、バンクチャパターン設定部620が設定するバンクチャパターンの探索方法について説明する。バンクチャパターンは、QC-LDPC符号の検査行列Hが、サブブロック行列を基本単位として構成されていることを利用して、バンクチャパターンを探索する。

10

【0115】

バンクチャパターン設定部620は、バンクチャパターンを探索する際、先ず、バンクチャパターンの周期を決定する。例えば、20ビットから送信しないビット（バンクチャビット）をKビット選択する場合、バンクチャパターンの周期は、20ビットとなる。なおこのとき、バンクチャパターンの周期の20ビット内に含まれる送信しないビット（バンクチャビット）の数は、K個とし、常に一定とする。

【0116】

本発明では、バンクチャパターンの周期を、QC-LDPC符号の検査行列の基本単位であるサブブロック行列 $I(p_j, 1)$ （q行r列（ $r = (q + p_j, 1) \bmod p$ （ $0 \leq q \leq p - 1$ ）が1であり、その他は“0”であるような巡回置換行列）の列数Lの整数倍、又は、列数Lの約数とする（式（1）参照）。

20

【0117】

例えば、図3に示すQC-LDPC符号の検査行列におけるサブブロック行列は、27行27列（ $L = 27$ ）の行列であるので、27の整数倍、又は、27の約数をバンクチャパターンの周期と設定し、送信しないビット（バンクチャビット）K個を設定することを提案する。

【0118】

一般に、ブロック符号では、ブロック長が長いほど良好な受信特性が得られる。しかし、ブロック長が長い場合には、ブロック長単位で、最良のバンクチャパターンを探索することは困難である。そのため、ブロック長が長い場合には、ランダムにバンクチャビットを選択する方式の採用が考えられる。しかし、この場合には、バンクチャ時の受信品質が大きく劣化する可能性がある。

30

【0119】

これに対し、バンクチャパターン設定部620が、QC-LDPC符号の検査行列Hを構成するサブブロック行列が規則的であることに着目し、サブブロック行列の列数の整数倍、又は、列数の約数ごとに、バンクチャパターンを探索する場合には、比較的短い時間で特性が良好となるバンクチャパターンを確実に見つけ出すことができる。

【0120】

バンクチャパターンの具体的な探索方法としては、例えば、所定のSNR（Signal-to-Noise power ratio：信号電力対雑音電力比）を設定し、バンクチャパターンごとに誤り率を求め、誤り率が低くなるバンクチャパターンを求めれば良い。

40

【0121】

送信装置は、このようにして探索されたバンクチャパターンを用いて、符号化系列をバンクチャすることにより、良好な受信品質を維持しつつ、伝送効率を向上させることができる。すなわち、図15の構成で重要な点は、バンクチャ部（データ削減部）630は、符号化系列をQC-LDPC符号の検査行列Hを構成するサブブロック行列の列数の整数倍、又は、列数の約数を単位として、バンクチャを行っている点である。

【0122】

一例として、バンクチャ部（データ削減部）630が、バンクチャパターンの周期をサ

50

サブブロック行列の列数 L とし、サブブロック行列の列数 L ごとに、送信しないビット（パンクチャビット）の数を K 個と一定とする場合について説明する。この場合、パンクチャ部（データ削減部）630は、パンクチャパターンを、QC-LDPC符号の検査行列 H を構成するサブブロック行列の列数の整数倍ごとに切り替える。

【0123】

パンクチャパターンの切り替え方法について、図16A～図16Cを用いて具体的に説明する。

【0124】

図16Aは、図3の検査行列 H に対し、サブブロック行列の列数（列数の1倍）ごとにパンクチャパターンを切り替える様子を示している。図3の検査行列 H は、27列のサブブロック行列から構成されるため、パンクチャ部（データ削減部）630は、 $x_1 \sim x_{27}$ に対しては、パンクチャパターン#0を用いて送信しないビット（パンクチャビット）を K 個選択する。また、パンクチャ部（データ削減部）630は、 $x_{28} \sim x_{54}$ に対してはパンクチャパターン#1を用いて、送信しないビット（パンクチャビット）を K 個選択する。また、パンクチャ部（データ削減部）630は、 $p_{622} \sim p_{648}$ に対しては、パンクチャパターン#23を用いて、送信しないビット（パンクチャビット）を K 個選択する。

【0125】

図16Bは、図3の検査行列 H に対し、サブブロック行列の列数の2倍ごとにパンクチャパターンを切り替える様子を示している。図3の検査行列 H は、27列のサブブロック行列から構成されるため、パンクチャ部（データ削減部）630は、 $x_1 \sim x_{27}$ 、 $x_{28} \sim x_{54}$ に対しては、パンクチャパターン#0を用いて送信しないビット（パンクチャビット）を K 個選択する。

【0126】

また、パンクチャ部（データ削減部）630は、 $x_{55} \sim x_{81}$ 、 $x_{82} \sim x_{108}$ に対してはパンクチャパターン#1を用いて、送信しないビット（パンクチャビット）を K 個選択する。また、パンクチャ部（データ削減部）630は、 $x_{109} \sim x_{135}$ 、 $x_{136} \sim x_{162}$ に対しては、パンクチャパターン#2を用いて、送信しないビット（パンクチャビット）を K 個選択する。

【0127】

図16Cは、図3の検査行列 H に対し、サブブロック行列の列数の約数9列を基本周期として、9列ごとにパンクチャパターンを切り替える様子を示している。具体的には、パンクチャ部（データ削減部）630は、 $x_1 \sim x_9$ に対しては、パンクチャパターン#0を用いて送信しないビット（パンクチャビット）を K 個選択する。

【0128】

パンクチャ部（データ削減部）630は、 $x_{10} \sim x_{18}$ に対しては、パンクチャパターン#1を用いて送信しないビット（パンクチャビット）を K 個選択する。パンクチャ部（データ削減部）630は、 $x_{19} \sim x_{27}$ に対しては、パンクチャパターン#2を用いて送信しないビット（パンクチャビット）を K 個選択する。

【0129】

同様に、パンクチャ部（データ削減部）630は、 $x_{28} \sim x_{36}$ に対しては、パンクチャパターン#3を用いて送信しないビット（パンクチャビット）を K 個選択する。パンクチャ部（データ削減部）630は、 $x_{37} \sim x_{45}$ に対しては、パンクチャパターン#4を用いて送信しないビット（パンクチャビット）を K 個選択する。パンクチャ部（データ削減部）630は、 $x_{46} \sim x_{54}$ に対しては、パンクチャパターン#5を用いて送信しないビット（パンクチャビット）を K 個選択する。

【0130】

同様に、パンクチャ部（データ削減部）630は、 $x_{622} \sim x_{630}$ に対しては、パンクチャパターン#69を用いて送信しないビット（パンクチャビット）を K 個選択する。パンクチャ部（データ削減部）630は、 $x_{631} \sim x_{639}$ に対しては、パンクチャ

10

20

30

40

50

パターン # 70 を用いて送信しないビット (バンクチャビット) を K 個選択する。バンクチャ部 (データ削減部) 630 は、 $x640 \sim x648$ に対しては、バンクチャパターン # 71 を用いて送信しないビット (バンクチャビット) を K 個選択する。

【0131】

なお、バンクチャ部 (データ削減部) 630 は、バンクチャパターン # 0 ~ # 2 から構成されるバンクチャパターン # $S0$ を定義し、 $x1 \sim x27$ に、バンクチャパターン # $S0$ を用いて送信しないビット (バンクチャビット) を $3K$ 個選択するようにしても良い。同様に、バンクチャ部 (データ削減部) 630 は、バンクチャパターン # 3 ~ # 5 から構成されるバンクチャパターン # $S1$ を定義し、 $x28 \sim x54$ に、バンクチャパターン # $S1$ を用いて送信しないビット (バンクチャビット) を $3K$ 個選択するようにしても良い。

10

【0132】

同様に、バンクチャ部 (データ削減部) 630 は、バンクチャパターン # 69 ~ # 71 から構成されるバンクチャパターン # $S23$ を定義し、 $x622 \sim x648$ に、バンクチャパターン # $S23$ を用いて送信しないビット (バンクチャビット) を $3K$ 個選択するようにしても良い。

【0133】

すなわち、サブブロック行列の列数の約数を基本周期としてバンクチャをすることは、QC-LDPC 符号の検査行列 H を構成するサブブロック行列の列数を単位 (周期) としてバンクチャすることと等価になる。

20

【0134】

以上のように、本実施の形態では、バンクチャパターン設定部 620 は、QC-LDPC 符号の検査行列 H を構成するサブブロック行列の列数の整数倍、又は、列数の約数ごとにバンクチャパターンを探索し、バンクチャ部 (データ削減部) 630 は、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の整数倍、又は、列数の約数ごとに、バンクチャパターンを切り替える。これにより、良好な受信品質が得られるバンクチャパターンを、比較的短い時間で確実に探索することができ、良好な受信品質を維持しつつ、伝送効率を向上させることができる。

【0135】

なお、以上の説明では、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の整数倍、又は、列数の約数ごとに、バンクチャパターンを切り替える場合を例に説明したが、バンクチャパターンを必ずしも切り替える必要はない。

30

【0136】

例えば、図 16A は、「バンクチャパターン # 0」、「バンクチャパターン # 1」、
・ ・ ・ 「バンクチャパターン # 23」が同一のバンクチャパターンであっても良い。また、
図 16B は、「バンクチャパターン # 0」、「バンクチャパターン # 1」、「バンクチャ
パターン # 2」、
・ ・ ・ が同一のバンクチャパターンであっても良い。

【0137】

また、図 16C は、「バンクチャパターン # 0」、「バンクチャパターン # 1」、
・ ・ ・ 「バンクチャパターン # 71」が同一のバンクチャパターンであっても良い。要するに
、バンクチャパターンの単位は、QC-LDPC 符号の検査行列を構成するサブブロック
行列の列数の整数倍、又は、列数の約数となっていれば良い。

40

【0138】

(実施の形態 4)

本実施の形態では、実施の形態 1 及び実施の形態 2 において説明した符号化方法を、制御情報に利用する場合の符号化方法の例を説明する。

【0139】

一例として、以下では、Coding rate(R)= $1/2$, LDPC code information block length(bits)=168, LDPC codeword block length(bits)=336 の QC-LDPC 符号を用い、200 ビットの制御情報を符号化する場合について説明する。

50

【 0 1 4 0 】

図 1 7 は、2 0 0 ビットの制御情報を 1 6 8 ビットと 3 2 ビットとに分割し、1 6 8 ビットをブロック # 1 に配置し、3 2 ビットをブロック # 2 に配置する場合を示している。図 1 7 において、ブロック # 2 には、ブロック長 1 6 8 ビットに対し、制御情報が 3 2 ビットしか配置されない。

【 0 1 4 1 】

以下、ブロック # 2 のように、送信する必要があるビットが、ブロック長より短いブロックは、実施の形態 2 において説明した特殊ブロックである。そのため、実施の形態 2 と同様に、ブロック # 2 では、情報ビットとして“ 0 ”が仮想ビットとして配置され、符号化される。この結果、ブロック # 1 とブロック # 2 とでは、受信品質にばらつきが発生し、結局、2 0 0 ビットの制御情報の受信品質は、受信品質が悪いブロックに依存してしまう。

10

【 0 1 4 2 】

そこで、本実施の形態では、図 1 8 に示すように、2 0 0 ビットの制御情報を、2 つのブロック # 1 , # 2 にできるだけ均等に配置し、各ブロックに対し実施の形態 1 で述べた符号化を行う。具体的には、制御情報が 2 0 0 ビットの場合、ブロック # 1 及びブロック # 2 の双方に、1 0 0 ビットずつ制御情報を配置する。

【 0 1 4 3 】

これにより、ブロック # 1 及びブロック # 2 が共に、特殊ブロックになるため、ブロック # 1 及びブロック # 2 の双方で、情報ビットとして“ 0 ”が仮想ビットとして配置されて、実施の形態 1 における符号化方法により符号化される。これにより、ブロック # 1 とブロック # 2 との受信品質が均等になり、通信相手に的確に伝送できるようになる。

20

【 0 1 4 4 】

なお、制御情報が 2 0 1 ビットの場合には、ブロック # 1 には制御情報を 1 0 1 ビット配置し、ブロック # 2 には制御情報を 1 0 0 ビット配置するようにする。この場合、ブロック # 1 における制御情報のビットの数とブロック # 2 における制御情報のビットの数との差は、高々 1 ビットである。このように、送信装置は、2 つのブロックに、送信する必要がある情報をできるだけ均等に配置することにより、各ブロック間の受信品質を均等にすることができるので、制御情報を通信相手に確実に伝送することができるようになる。

【 0 1 4 5 】

30

以上のように、本実施の形態では、制御情報を複数のブロックにできるだけ均等に配置するようにした。これにより、送信装置は、配置後の各ブロックに対し、実施の形態 1 において説明した符号化方法を適用することにより、制御情報等の通信確立に必要な情報を確実に通信相手に伝送することができる。

【 0 1 4 6 】

なお、本実施の形態における特殊ブロックの生成方法は、実施の形態 2 で説明した特殊ブロックの生成方法と同様である。つまり、送信装置は、送信する必要のない情報ビット及びパリティビットの双方を送信しないビットと設定（バンクチャビットと設定）することになる。

【 0 1 4 7 】

40

（実施の形態 5）

本実施の形態では、QC - LDPC 符号の一例を示すとともに、当該 QC - LDPC 符号に最適なバンクチャパターンについて説明する。

【 0 1 4 8 】

QC - LDPC 符号の検査行列 H は、式（ 4 ）のように定義する。

【数 4】

$$H = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & \cdots & P_{0,n_b-2} & P_{0,n_b-1} \\ P_{1,0} & P_{1,1} & P_{1,2} & \cdots & P_{1,n_b-2} & P_{1,n_b-1} \\ P_{2,0} & P_{2,1} & P_{2,2} & \cdots & P_{2,n_b-2} & P_{2,n_b-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ P_{m_b-1,0} & P_{m_b-1,1} & P_{m_b-1,2} & \cdots & P_{m_b-1,n_b-2} & P_{m_b-1,n_b-1} \end{bmatrix} = P^{H_b} \quad \cdots (4)$$

【0149】

10

式(4)の検査行列Hは、m行n列の行列である。ここで、nは、符号長であり、mは、パリティビット数である。従って、システムティックビット数kは、 $k = n - m$ となる。また、式(4)において、 $P_{i,j}$ は、z行z列の巡回置換行列又はz行z列のゼロ行列である。

【0150】

ここで、式(4)の検査行列Hは、 n_b 行 m_b 列の行列 H_b により展開する。なお、 $m = z \times m_b$ 、 $n = z \times n_b$ の関係が成り立つ。また、行列 H_b の各要素は、 $P_{i,j}$ において要素が“1”の場合、“1”とし、 $P_{i,j}$ において要素が“0”の場合、“0”とする。

【0151】

20

ここで、 $P_{i,j}$ は、巡回置換行列として、z行z列の単位行列、又は、z行z列の単位行列を巡回シフトした行列の集合である。巡回置換行列は、単位行列、又は、単位行列を巡回シフトした行列の集合であるので、行列 H_b を、行列 H_b と大きさが同一の行列 H_{b_m} に分解すると、行列 H_{b_m} は、ゼロ行列、又は、単位行列を巡回シフトした行列で示される。

【0152】

なお、以降、行列 H_{b_m} において、ゼロ行列は“-1”と標記する。また、単位行列は“0”と標記する。また、単位行列の巡回置換行列は、その巡回シフト量 $p(i,j)$ (> 0)を用いて“ $p(i,j)$ ”と標記する。このようにコンパクトに標記された行列 H_{b_m} の集合として、行列 H_b を表現することができる。

30

【0153】

ところで、行列 H_b は、式(5)に示すように、2つのサブ行列 H_{b_1} 、 H_{b_2} に分けられる。サブ行列 H_{b_1} は、情報ビットに関連する部分行列であり、サブ行列 H_{b_2} は、パリティビットに関連する部分行列である。

【数 5】

$$H_b = [(H_{b1})_{mb \times kb} | (H_{b2})_{mb \times mb}] = 0 \quad \cdots (5)$$

【0154】

サブ行列 H_{b_2} は、式(6)に示すように、更に、ベクトル h_b とサブ行列 H'_{b_2} に分けられる。

40

【数 6】

$$H_{b2} = [h_b | H'_{b2}] = \begin{bmatrix} h_b(0) & | & 1 & & & \\ h_b(1) & | & 1 & 1 & & 0 \\ \cdots & | & & 1 & \ddots & \\ \cdots & | & & & \ddots & 1 \\ \cdots & | & 0 & & 1 & 1 \\ h_b(m_b-1) & | & & & & 1 \end{bmatrix} \quad \cdots (6)$$

10

【0 1 5 5】

式(6)において、サブ行列 H'_{b2} は、 i 行 j 列 ($i = j$ 及び $i = j + 1$) の部分が“1”であり、他の部分は“0”の行列である。サブ行列 H'_{b2} において、“1”と標記される部分は、単位行列のシフト量が0であることを示す。つまり、サブ行列 H'_{b2} は、行列 H_b に展開される際、 z 行 z 列の単位行列によって置き換えられる。

【0 1 5 6】

また、ベクトル h_b の一番上 ($h_b(0)$) と一番下 ($h_b(m_b - 1)$) とには、同じ巡回シフト量が割り当てられるとする。

【0 1 5 7】

以下では、式(7)により定義される行列 H_b を考える。式(7)により定義される検査行列 H は、各符号化率における最大符号長に対応することができる。

20

【数 7】

$$p(f, i, j) = \begin{cases} p(i, j) & , \quad p(i, j) \leq 0 \\ \left\lfloor p(i, j) \cdot \frac{z_f}{96} \right\rfloor & , \quad p(i, j) > 0 \cdots (7) \end{cases}$$

ここで、 $\left\lfloor p(i, j) \cdot \frac{z_f}{96} \right\rfloor$ は、 $p(i, j) \cdot \frac{z_f}{96}$ の整数部分を示す。

30

【0 1 5 8】

式(7)において、 $p(f, i, j)$ は、単位行列の巡回シフト量を示し、 f は、各符号化率に対応する符号長のインデックスを示す。また、 z_f は、展開ファクタと呼ばれ、 $z_f = k / n$ の関係がある。

【0 1 5 9】

式(7)に基づく符号化率 $1/2 (= k/n)$ の行列 H_b を、式(8)に示す。

【数 8】

$$\begin{bmatrix} -1 & 94 & 73 & -1 & -1 & -1 & -1 & -1 & 55 & 83 & -1 & -1 & 7 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 27 & -1 & -1 & -1 & 22 & 79 & 9 & -1 & -1 & -1 & 12 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 24 & 22 & 81 & -1 & 33 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ 61 & -1 & 47 & -1 & -1 & -1 & -1 & -1 & 65 & 25 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 39 & -1 & -1 & -1 & 84 & -1 & -1 & 41 & 72 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 46 & 40 & -1 & 82 & -1 & -1 & -1 & 79 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\ -1 & -1 & 95 & 53 & -1 & -1 & -1 & -1 & -1 & 14 & 18 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\ -1 & 11 & 73 & -1 & -1 & -1 & 2 & -1 & -1 & 47 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\ 12 & -1 & -1 & -1 & 83 & 24 & -1 & 43 & -1 & -1 & -1 & 51 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 94 & -1 & 59 & -1 & -1 & 70 & 72 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & 7 & 65 & -1 & -1 & -1 & -1 & 39 & 49 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\ 43 & -1 & -1 & -1 & -1 & 66 & -1 & 41 & -1 & -1 & -1 & 26 & 7 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 \end{bmatrix}$$

40

… (8)

50

【 0 1 6 0 】

式 (8) において、“ 0 ” は、単位行列を示す。また、“ - 1 ” は、ゼロ行列を示す。また、たとえば、1 行 2 列目の “ 9 4 ” は、単位行列を 9 4 だけサイクリックシフトした行列を示す。同様に、4 行 1 列目の “ 6 1 ” は、単位行列を 6 1 だけサイクリックシフトした行列を示す。

【 0 1 6 1 】

また、式 (7) に基づく符号化率 $5 / 6$ ($= k / n$) の行列 H_b を、式 (9) に示す。

【 数 9 】

1	25	55	-1	47	4	-1	91	84	8	86	52	82	33	5	0	36	20	4	77	80	0	-1	-1
-1	6	-1	36	40	47	12	79	47	-1	41	21	12	71	14	72	0	44	49	0	0	0	0	-1
51	81	83	4	67	-1	21	-1	31	24	91	61	81	9	86	78	60	88	67	15	-1	-1	0	0
50	-1	50	15	-1	36	13	10	11	20	53	90	29	92	57	30	84	92	11	66	80	-1	-1	0

10

… (9)

【 0 1 6 2 】

以上、符号化率 $1 / 2$ 及び $5 / 6$ の QC - L D P C の行列 H_b の一例を示した。以下、これら QC - L D P C の行列 H_b に適用可能なパンクチャパターンについて説明する。

【 0 1 6 3 】

図 1 9 A には、式 (8) に示した符号化率 $1 / 2$ の QC - L D P C の行列 H_b を示す。図 1 9 A に示されるように、符号化率 $1 / 2$ の行列 H_b は、情報ビットに関連する部分行列 H_{b_1} が 1 2 行であるため、パリティビットに関連する部分行列 H_{b_2} は 1 2 列となる。

20

【 0 1 6 4 】

図 1 9 A のパリティビットに関連する部分行列 H_{b_2} は、1 行 1 列目及び 1 2 行 1 列目を除き、“ - 1 ” , “ 0 ” で構成されており、規則的な配列となっている。上述したように、“ - 1 ” は、ゼロ行列を示し、“ 0 ” は、単位行列を示す。また、1 行 1 列目及び 1 2 行 1 列目の “ 7 ” は、単位行列を 7 だけサイクリックシフトした巡回置換行列である。

【 0 1 6 5 】

このとき、パリティビットに関連する部分行列 H_{b_2} の列において、単位行列、ゼロ行列により構成される部分は、同一のパンクチャパターンを用いるようにしても、受信品質に与える影響が低い。そのため、単位行列、ゼロ行列により構成される部分については、同一のパンクチャパターン # A を用いても、良好な受信特性を得ることができる (図 1 9 A 参照) 。なお、単位行列、ゼロ行列により構成される部分に該当しない列については、異なるパンクチャパターンと設定するものとする。ただし、一部又は全てのパンクチャパターンが同一のパンクチャパターンとなっても良い。

30

【 0 1 6 6 】

更に、符号化系列は、実施の形態 3 で述べたパンクチャ方法と組み合わせることも可能である。つまり、符号化系列は、QC - L D P C 符号の検査行列 H を構成するサブブロック行列の列数の整数倍、又は、列数の約数を単位として、パンクチャを行うと更に効果的である。図 1 9 B、図 1 9 C では、パリティビットに関連する部分行列 H_{b_2} に対し、QC - L D P C 符号の検査行列 H を構成するサブブロック行列の列数の整数倍、又は、列数の約数を単位として、パンクチャを行う場合の例を示している。

40

【 0 1 6 7 】

図 1 9 B には、式 (8) に示した符号化率 $1 / 2$ の QC - L D P C の行列 H_b 及びパンクチャパターンの別の適用例について示す。図 1 9 B は、QC - L D P C 符号の検査行列を構成するサブブロック行列の列数の整数倍 (2 倍) に、パンクチャパターンの周期を設定した場合の例である。なお、図 1 9 B は、単位行列、ゼロ行列により構成される部分には、同一のパンクチャパターン # B を用いるようにした例である。

【 0 1 6 8 】

また、図 1 9 C には、式 (8) に示した符号化率 $1 / 2$ の QC - L D P C の行列 H_b 及

50

びパンクチャパターンの別の適用例について示す。図 19C は、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の $1/2$ ごとにパンクチャパターンを生成した場合の例である。なお、図 19C は、単位行列、ゼロ行列により構成される部分には、同一のパンクチャパターンを用いるようにした例である。

【0169】

具体的には、図 19C は、100 行 100 列のサブブロック行列から構成される検査行列 H に対し、サブブロック行列の列数の $1/2$ である約 50 列を基本周期として、50 列ごとにパンクチャパターンを切り替える様子を示している。

【0170】

具体的には、パンクチャ部（データ削減部）630 は、p100 ~ p149 に対しては、パンクチャパターン #1 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）630 は、p150 ~ p199 に対しては、パンクチャパターン #2 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）630 は、p200 ~ p249 に対しては、パンクチャパターン #3 を用いて送信しないビット（パンクチャビット）を選択する。

【0171】

パンクチャ部（データ削減部）630 は、p250 ~ p299 に対しては、パンクチャパターン #4 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）630 は、p1100 ~ p1149 に対しては、パンクチャパターン #21 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）630 は、p1150 ~ p1199 に対しては、パンクチャパターン #22 を用いて送信しないビット（パンクチャビット）を選択する。

【0172】

図 20A は、式 (9) に示した符号化率 $5/6$ の QC-LDPC の行列 H_b を示す。図 20A に示されるように、符号化率 $5/6$ の検査行列 H_b は、情報ビットに関連する部分行列 $H_{b,1}$ が 4 行であるため、パリティビットに関連する部分行列 $H_{b,2}$ は 4 列となる。

【0173】

図 20A のパリティビットに関連する部分行列 $H_{b,2}$ は、1 行 1 列目及び 4 行 1 列目を除き、“-1”，“0”で構成されており、規則的な配列となっている。また、1 行 1 列目及び 4 行 1 列目の“80”は、単位行列を 80 だけサイクリックシフトした巡回置換行列である。

【0174】

このように、符号化率 $5/6$ の場合にも、パリティビットに関連する部分行列 $H_{b,2}$ の列において、単位行列、ゼロ行列により構成される部分は、同一のパンクチャパターンを用いるようにしても、受信品質に与える影響が低い。そのため、単位行列、ゼロ行列により構成される部分の列については、同一のパンクチャパターン #A を用いても、受信装置は、良好な受信特性を得ることができる（図 20A 参照）。なお、単位行列、ゼロ行列により構成される部分に該当しない列については、異なるパンクチャパターンと設定するものとするが、一部が同一のパンクチャパターンとなっても良い。

【0175】

図 20B には、式 (9) に示した符号化率 $5/6$ の QC-LDPC の行列 H_b 及びパンクチャパターンの別の適用例について示す。図 20B は、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の整数倍（3 倍）に、パンクチャパターンの周期を設定した場合の例である。

【0176】

また、図 20C には、式 (9) に示した符号化率 $5/6$ の QC-LDPC の行列 H_b 及びパンクチャパターンの別の適用例について示す。図 20C は、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の $1/2$ ごとにパンクチャパターンを生成した場合の例である。なお、図 20B と同様に、図 20C は、単位行列、ゼロ行列により構成される部分には、同一のパンクチャパターンを用いるようにした例である。

【 0 1 7 7 】

具体的には、図 2 0 C は、1 0 0 行 1 0 0 列のサブブロック行列から構成される検査行列 H に対し、サブブロック行列の列数の $1/2$ である約数 5 0 列を基本周期として、5 0 列ごとにパンクチャパターンを切り替える様子を示している。

【 0 1 7 8 】

具体的には、パンクチャ部（データ削減部）6 3 0 は、p 1 0 0 ~ p 1 4 9 に対しては、パンクチャパターン # 1 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）6 3 0 は、p 1 5 0 ~ p 1 9 9 に対しては、パンクチャパターン # 2 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）6 3 0 は、p 2 0 0 ~ p 2 4 9 に対しては、パンクチャパターン # 3 を用いて送信しないビット（パンクチャビット）を選択する。

10

【 0 1 7 9 】

パンクチャ部（データ削減部）6 3 0 は、p 2 5 0 ~ p 2 9 9 に対しては、パンクチャパターン # 4 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）6 3 0 は、p 3 0 0 ~ p 3 4 9 に対しては、パンクチャパターン # 5 を用いて送信しないビット（パンクチャビット）を選択する。パンクチャ部（データ削減部）6 3 0 は、p 3 5 0 ~ p 3 9 9 に対しては、パンクチャパターン # 6 を用いて送信しないビット（パンクチャビット）を選択する。

【 0 1 8 0 】

このように、パリティビットに関連する部分行列 H_{b_2} の列において、単位行列、ゼロ行列により構成される部分は、同一のパンクチャパターンを設定し、単位行列、ゼロ行列により構成される部分に該当しない列については、異なるパンクチャパターンと設定する。

20

【 0 1 8 1 】

なお、単位行列、ゼロ行列により構成される部分に該当しない列は、例えば、実施の形態 3 で述べたように、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の整数倍、又は、列数の約数ごとに、パンクチャパターンを切り替えるようにしても良い。

【 0 1 8 2 】

また、単位行列、ゼロ行列により構成される部分に該当しない列は、パンクチャパターンのパターン長が、QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の整数倍、又は、列数の約数ごとである同一のパンクチャパターンを適用するようにしても良い。

30

【 0 1 8 3 】

（実施の形態 6）

実施の形態 5 で説明した QC-LDPC 符号を用い、かつ、実施の形態 4 で説明した QC-LDPC 符号の検査行列を構成するサブブロック行列の列数の整数倍、又は、列数の約数を単位として、パンクチャを行っていて、全てにおいて、同一のパンクチャパターンを用いたときの例を示す。

【 0 1 8 4 】

実施の形態 6 は、符号化率 $1/2$ の式 (8) の検査行列をもつ QC-LDPC 符号からパンクチャにより、符号化率約 0.65 を実現するためのパンクチャパターンについて説明する。ただし、QC-LDPC 符号の検査行列を構成するサブブロック行列のサイズは、行数 3 5 0、列数 3 5 0 とする。したがって、QC-LDPC 符号の Information block length(bits) は 4200 となり、LDPC codeword block length(bits) は 8400 となる。

40

【 0 1 8 5 】

このとき、LDPC 符号の codeword を以下のようにあらわす。

$v = [x_0, x_1, \dots, x_{4198}, x_{4199}, p_0, p_1, \dots, p_{4198}, p_{4199}]$
 $= [s_0, s_1, s_2, \dots, s_{8397}, s_{8398}, s_{8399}]$
 $= [v_0, v_1, v_2, \dots, v_{167}]$

50

ただし、 v はcodeword、 x は情報、 p はパリティを意味する。

【 0 1 8 6 】

なお、 $v_0, v_1, \dots, v_i, \dots, v_{167}$ は以下のようにあらわされる。

$v_0=[s_0, s_1, \dots, s_{48}, s_{49}]$ 、
 $v_1=[s_{50}, s_{51}, \dots, s_{98}, s_{99}]$ 、 \dots 、
 $v_i=[s_{50*i}, s_{50*i+1}, \dots, s_{50*i+48}, s_{50*i+49}]$ 、 \dots 、
 $v_{167}=[s_{8350}, s_{8351}, \dots, s_{8398}, s_{8399}]$

【 0 1 8 7 】

本発明者らは、パンクチャパターンを探索した結果、QC-LDPC符号の検査行列を構成するサブブロック行列の列数の約数である50をパンクチャパターンの周期とすると良好な受信品質を与えることを確認した。

10

【 0 1 8 8 】

良好な受信品質を与えるパンクチャリングパターンは以下のとおりである。

(1, 8, 19, 20, 25, 28, 29, 31, 38, 40, 41)

【 0 1 8 9 】

別の表現として、パンクチャテーブル w は、

$w=[1011111101 \ 1111111110 \ 0111101100 \ 1011111101 \ 0011111111]$

であらわされる。

【 0 1 9 0 】

このとき、 w に含まれる0が送信しないビットを意味する。つまり、パンクチャテーブル w は、 v_i に対して、図21に示すように送信しないビットを決定する。したがって、 $v_i=[s_{50*i}, s_{50*i+1}, \dots, s_{50*i+48}, s_{50*i+49}]$ に対して、送信しないビットを除いた、送信するデータビット v_i' は、

20

$v_i'=[s_{50*i}, s_{50*i+2}, s_{50*i+3}, s_{50*i+4}, s_{50*i+5}, s_{50*i+6}, s_{50*i+7}, s_{50*i+9}, s_{50*i+10}, s_{50*i+11}, s_{50*i+12}, s_{50*i+13}, s_{50*i+14}, s_{50*i+15}, s_{50*i+16}, s_{50*i+17}, s_{50*i+18}, s_{50*i+21}, s_{50*i+22}, s_{50*i+23}, s_{50*i+24}, s_{50*i+26}, s_{50*i+27}, s_{50*i+30}, s_{50*i+32}, s_{50*i+33}, s_{50*i+34}, s_{50*i+35}, s_{50*i+36}, s_{50*i+37}, s_{50*i+39}, s_{50*i+42}, s_{50*i+43}, s_{50*i+44}, s_{50*i+45}, s_{50*i+46}, s_{50*i+47}, s_{50*i+48}, s_{50*i+49}]$ とあらわされる。

【 0 1 9 1 】

符号化率5/6の式(9)の検査行列をもつQC-LDPC符号から、パンクチャにより、符号化率約0.95を実現するためのパンクチャパターンについて説明する。ただし、QC-LDPC符号の検査行列を構成するサブブロック行列のサイズは、行数210、列数210とする。したがって、QC-LDPC符号において、Information block length(bits)は4200となり、LDPC codeword block length(bits)は5040となる。

30

【 0 1 9 2 】

このとき、LDPC符号のcodewordを以下のようにあらわす。

$v=[x_0, x_1, \dots, x_{4198}, x_{4199}, p_0, p_1, \dots, p_{838}, p_{839}]$
 $=[s_0, s_1, s_2, \dots, s_{5037}, s_{5038}, s_{5039}]$
 $=[v_0, v_1, v_2, \dots, v_{79}]$

40

ただし、 v はcodeword、 x は情報、 p はパリティを意味する。

【 0 1 9 3 】

なお、 $v_0, v_1, \dots, v_i, \dots, v_{79}$ は以下のようにあらわされる。

$v_0=[s_0, s_1, \dots, s_{61}, s_{62}]$ 、
 $v_1=[s_{63}, s_{64}, \dots, s_{124}, s_{125}]$ 、 \dots 、
 $v_i=[s_{63*i}, s_{63*i+1}, \dots, s_{63*i+61}, s_{63*i+62}]$ 、 \dots 、
 $v_{79}=[s_{4977}, s_{4978}, \dots, s_{5038}, s_{5039}]$ 。

【 0 1 9 4 】

本発明者らは、パンクチャパターンを探索した結果、63をパンクチャパターンの周期とすると良好な受信品質を与えることを確認した。

50

【 0 1 9 5 】

良好な受信品質を与えるパンクチャリングパターンは以下のとおりである。

(3, 18, 20, 27, 39, 50, 60)

【 0 1 9 6 】

別の表現として、パンクチャテーブルwは

$w = [1110111111 \ 1111111101 \ 0111111011 \ 1111111110 \ 1111111111 \ 0111111111 \ 0111]$

であらわされる。

【 0 1 9 7 】

このとき、wに含まれる0が送信しないビットを意味する。つまり、パンクチャテーブルwは、 v_i に対して、図22に示すように送信しないビットを決定する。したがって、 $v_i = [s_{63 \cdot i}, s_{63 \cdot i + 1}, \dots, s_{63 \cdot i + 61}, s_{63 \cdot i + 62}]$ に対して送信しないビットを除いた、送信するデータビット v_i' は、

$v_i' = [s_{63 \cdot i}, s_{63 \cdot i + 1}, s_{63 \cdot i + 2}, s_{63 \cdot i + 4}, s_{63 \cdot i + 5}, s_{63 \cdot i + 6}, s_{63 \cdot i + 7}, s_{63 \cdot i + 8}, s_{63 \cdot i + 9}, s_{63 \cdot i + 10}, s_{63 \cdot i + 11}, s_{63 \cdot i + 12}, s_{63 \cdot i + 13}, s_{63 \cdot i + 14}, s_{63 \cdot i + 15}, s_{63 \cdot i + 16}, s_{63 \cdot i + 17}, s_{63 \cdot i + 19}, s_{63 \cdot i + 21}, s_{63 \cdot i + 22}, s_{63 \cdot i + 23}, s_{63 \cdot i + 24}, s_{63 \cdot i + 25}, s_{63 \cdot i + 26}, s_{63 \cdot i + 28}, s_{63 \cdot i + 29}, s_{63 \cdot i + 30}, s_{63 \cdot i + 31}, s_{63 \cdot i + 32}, s_{63 \cdot i + 33}, s_{63 \cdot i + 34}, s_{63 \cdot i + 35}, s_{63 \cdot i + 36}, s_{63 \cdot i + 37}, s_{63 \cdot i + 38}, s_{63 \cdot i + 40}, s_{63 \cdot i + 41}, s_{63 \cdot i + 42}, s_{63 \cdot i + 43}, s_{63 \cdot i + 44}, s_{63 \cdot i + 45}, s_{63 \cdot i + 46}, s_{63 \cdot i + 47}, s_{63 \cdot i + 48}, s_{63 \cdot i + 49}, s_{63 \cdot i + 51}, s_{63 \cdot i + 52}, s_{63 \cdot i + 53}, s_{63 \cdot i + 54}, s_{63 \cdot i + 55}, s_{63 \cdot i + 56}, s_{63 \cdot i + 57}, s_{63 \cdot i + 58}, s_{63 \cdot i + 59}, s_{63 \cdot i + 61}, s_{63 \cdot i + 62}]$ とあらわされる。

【 0 1 9 8 】

このとき、パンクチャパターンの周期を20～90程度とすると、受信時のデータ品質が良好となる。なお、パンクチャパターンの周期とは、パンクチャパターンの最小周期をいう。例えば、パンクチャテーブル $w_1 = [001]$ のパンクチャパターンの周期は3である。又、パンクチャテーブル $w_2 = [001001]$ は、周期6の構成であるが、2つのパンクチャテーブル $w_1 = [001]$ から構成され、パンクチャテーブル w_1 のパンクチャパターンの周期は3であるため、パンクチャテーブル w_2 のパンクチャパターンの（最小）周期は、パンクチャテーブル w_1 と同様に3である。すなわち、パンクチャパターンの周期とは、パンクチャパターンを構成するパターンのうち、最小パターンのパターン長をいう。また、パンクチャテーブル $w_3 = [010]$ は、 w_1 をサイクリックシフトしたものと同一となるが、上記のw, v_i , v_i' の関係を考慮すると、 w_3 と w_1 は異なるパンクチャパターンといえることができる。すなわち、パンクチャテーブル w_x とパンクチャテーブル w_y があったとき、たとえ w_x をサイクリックシフト（ただし、0ビットサイクリックシフトを除く）して w_y と同一となっても w_x と w_y は異なるパンクチャパターンである。

【 0 1 9 9 】

パンクチャパターンの周期が長すぎると、送信しないビット（パンクチャビット）の配置にランダム性が発生し、バイナリ消失チャネルにおいて、ランダムエラーが発生したモデルに近づくため、受信時のデータ品質は悪くなる。一方、パンクチャパターンの周期が短すぎると、送信しないビット（パンクチャビット）の配置が片寄るため、適切なパンクチャパターンとなる可能性が低くなり、受信時のデータ品質は悪くなる。このため、パンクチャパターンの周期を20～90程度とすることが重要となる。

【 0 2 0 0 】

また、パンクチャパターンの周期を20～90程度とした際、パンクチャテーブルwに0が3つ以上含まれるようにすると、受信時のデータ品質は良好となる（受信（復号）時に良好なデータ品質を得ることができるパンクチャパターンを生成できる可能性が高くなる）。パンクチャテーブルwに0が3つ以上含まれると、送信しないビット（パンクチャビット）の配置に規則性がなくなりランダム性が増すため、受信時のデータ品質が良好となる。

【 0 2 0 1 】

更に、パンクチャパターンの周期を20～90程度とし、かつ、パンクチャテーブルw

に0が3つ以上含まれるようにし、かつ、QC-LDPC符号の検査行列を構成するサブブロック行列の列数の整数倍、又は、列数の約数をパンクチャパターンの周期とすると、受信（復号）時に良好なデータ品質を得ることができるパンクチャパターンを生成できる可能性が高くなる。

【0202】

上記以外におけるパンクチャパターンは以下のとおりである。

QC-LDPC符号の検査行列を構成するサブブロック行列のサイズは、行数80、列数80とし、符号化率1/2の式(8)の検査行列をもつQC-LDPC符号(Information block length(bits)=960, LDPC codeword block length(bits)=1920)からパンクチャにより、符号化率約0.65、0.75を実現するためのパンクチャパターンは以下のとおりである。

10

符号化率約0.65の場合：w=[1111110110 0100111111]

符号化率約0.75の場合：w=[1100111111 1101111110 0111110001 1110000111]

【0203】

QC-LDPC符号の検査行列を構成するサブブロック行列のサイズは、行数48、列数48とし、符号化率5/6の式(9)の検査行列をもつQC-LDPC符号(Information block length(bits)=960, LDPC codeword block length(bits)=1152)から、パンクチャにより、符号化率約0.95を実現するためのパンクチャパターンは以下のとおりである。

w=[1111111110 1111111111 0111101111 1111001111 111011111]

20

【0204】

QC-LDPC符号の検査行列を構成するサブブロック行列のサイズは、行数180、列数180とし、符号化率1/2の式(8)の検査行列をもつQC-LDPC符号(Information block length(bits)=2160, LDPC codeword block length(bits)=4320)からパンクチャにより、符号化率約0.65、0.75を実現するためのパンクチャパターンは以下のとおりである。

符号化率約0.65の場合：w=[1011111100 0011111101 1111100111 0111111]

符号化率約0.75の場合：w=[1111110100 0001101001 1111111110]

【0205】

QC-LDPC符号の検査行列を構成するサブブロック行列のサイズは、行数108、列数108とし、符号化率5/6の式(9)の検査行列をもつQC-LDPC符号(Information block length(bits)=2160, LDPC codeword block length(bits)=2592)から、パンクチャにより、符号化率約0.95を実現するためのパンクチャパターンは以下のとおりである。

30

w=[1011111111 1111011111 11101111]

【0206】

(実施の形態7)

実施の形態5では、パリティ検査行列 H_b において、単位行列及びゼロ行列により構成されるサブ行列 H'_{b_2} (式(6)参照)とサブ行列 H'_{b_2} 以外のサブ行列(以下「 H'_{b_1} ($=H_{b_1}+h_b$)」)と標記する。式(5)及び(6)参照)とで、異なるパンクチャパターンを用いる場合について説明した。その一例として、実施の形態5では、図19A~図19C、図20A~図20Cに示すように、単位行列及びゼロ行列により構成されるサブ行列 H'_{b_2} に、サブブロック行列の列数の整数倍、又は、列数の約数を単位として、同一のパンクチャパターンを用いる場合について説明した。

40

【0207】

本実施の形態では、実施の形態5と同様に、パリティ検査行列 H_b において、単位行列及びゼロ行列により構成されるサブ行列 H'_{b_2} とサブ行列 H'_{b_1} とで、異なるパンクチャパターンを用いる場合について説明する。具体的には、図20Bに対応する図23に示すように、サブ行列 H'_{b_1} には、サブ行列 H'_{b_1} の列数をパンクチャ周期とするパンクチャパターン#p1を用い、サブ行列 H'_{b_2} には、サブ行列 H'_{b_2} の列数をパン

50

クチャパターン周期とするパンクチャパターン # p 2 を用いて、符号化率 2 0 / 2 1 を実現する場合について説明する。

【 0 2 0 8 】

以下では、一例として、符号化率 5 / 6 の式 (9) の検査行列をもつ Q C - L D P C 符号から、パンクチャにより、符号化率 2 0 / 2 1 を実現するためのパンクチャパターンについて説明する。

【 0 2 0 9 】

図 2 4 の検査行列 H_b は、式 (9) に示した符号化率 5 / 6 の Q C - L D P C の検査行列 H_b である。式 (9) の検査行列 H_b は、4 行 2 4 列のサブブロック行列から構成される。なお、以下では、Q C - L D P C 符号の検査行列を構成するサブブロック行列のサイズが、行数 4 8、列数 4 8 とする。したがって、Q C - L D P C 符号において、Information block length(bits)は960となり、LDPC codeword block length(bits)は1152となる。

10

【 0 2 1 0 】

このとき、L D P C 符号のcodewordを以下のようにあらわす。

$v=[x_0, x_1, \dots, x_{958}, x_{959}, p_0, p_1, \dots, p_{190}, p_{191}]$
 $=[s_0, s_1, s_2, \dots, s_{1149}, s_{1150}, s_{1151}]$
 $=[v_0, v_1, v_2, \dots, v_{24}]$

ただし、vはcodeword、xは情報、pはパリティを意味する。

【 0 2 1 1 】

20

なお、 $v_0, v_1, \dots, v_i, \dots, v_{23}$ は以下のようにあらわされる。

$v_0=[s_0, s_1, \dots, s_{46}, s_{47}]$ 、
 $v_1=[s_{48}, s_{49}, \dots, s_{94}, s_{95}]$ 、 \dots 、
 $v_i=[s_{48*i}, s_{48*i+1}, \dots, s_{48*i+46}, s_{48*i+47}]$ 、 \dots 、
 $v_{23}=[s_{1104}, s_{1105}, \dots, s_{1150}, s_{1151}]$ 。

【 0 2 1 2 】

図 2 4 において、# 0 は、 x_0, x_1, \dots, x_{47} に対応する部分行列を示し、# 1 は、 $x_{48}, x_{49}, \dots, x_{95}$ に対応する部分行列を示す。また、# 2 1 は、 $p_{48}, p_{49}, \dots, p_{95}$ に対応する部分行列を示し、# 2 2 は、 $p_{96}, p_{97}, \dots, p_{143}$ に対応する部分行列を示し、# 2 3 は、 $p_{144}, p_{145}, \dots, p_{191}$ に対応する部分行列を示す。

30

【 0 2 1 3 】

図 2 4 において、サブ行列 H'_{b_1} は、# 0 ~ 2 0 により構成され、サブ行列 H'_{b_2} は、# 2 1, # 2 2, # 2 3 により構成される。# 2 1, # 2 2, # 2 3 は、単位行列 (“ 0 ”) 及びゼロ行列 (“ - 1 ”) により構成される。このように、式 (9) に示す Q C - L D P C 符号の検査行列 H_b は、単位行列及びゼロ行列で構成されるサブ行列 H'_{b_2} を含んでいる。

【 0 2 1 4 】

本実施の形態では、サブ行列 H'_{b_2} 及び B P 復号の特徴を考慮して最適なパンクチャパターンを以下のようにして決定する。

40

【 0 2 1 5 】

B P 復号では、行演算と列演算とを反復して各ビットの対数尤度比を得る。

【 0 2 1 6 】

B P 復号の行演算では、対数尤度比が更新される。このとき、送信されなかった (パンクチャ) ビットは、復号時には、消失ビットとして扱われることになり、消失ビットに対しては、初期の対数尤度比が存在しないため、対数尤度比が 0 に設定される。初期の対数尤度比が存在しない消失ビットが、同一行に 2 つ以上含まれると、当該行では、列演算により消失ビットの対数尤度比が更新されるまで、行演算単独では対数尤度比が更新されない。そのため、消失ビットは同一行において 2 ビット未満であるのが好ましい。

【 0 2 1 7 】

50

B P 復号の列演算では、外部値が更新される。消失ビットの外部値は、同一列の自身を除く“1”の対数尤度比の加算結果により更新される。したがって、列重みが大きい場合には、消失ビットの外部値は、同一列の自身を除く複数の“1”の対数尤度比の加算結果により更新されるので、外部値における対数尤度比の絶対値は大きくなり、これにより、対数尤度比が収束しやすい。一方、列重みが小さい場合には、加算される対数尤度比の数が少ないため、外部値における対数尤度比の絶対値は大きくなりづらく、これにより、対数尤度比が収束しにくいという性質をもつことになる。

【0218】

特に、列重みが2の場合には、検査行列における列重み2に相当する2つの“1”において、外部値が単に交換されるだけとなるため、対数尤度比の絶対値は大きくなりづらく、反復処理を何度行っても信頼度が伝播されず、受信品質の劣化を招く原因となる。したがって、外部値の大きさを適切に更新するためには、消失ビットの列重みが3以上であるのが好ましい。

10

【0219】

このように、B P 復号の特徴を考慮すると、行演算の観点からは、1) 同一行において消失ビットが2ビット未満となるのが好ましく、また、列演算の観点からは、2) 消失ビットの列重みが3以上であるのが好ましい。

【0220】

本実施の形態では、上記1)及び2)を考慮してバンクチャパターンを設定する。なお、以下では、サブブロック行列の列数を単位として符号化系列をバンクチャする場合を例に説明する。

20

【0221】

式(9)のパリティ検査行列 H_b を、サブブロック行列を1単位として表現したとき、サブ行列 H'_{b_2} では、 j 列の i 行及び $(i+1)$ 行には単位行列(“0”)が配置され、 j 列の i 行及び $(i+1)$ 行以外の行にはゼロ行列(“-1”)が配置され、かつ、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行には単位行列(“0”)が配置され、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行以外の行にはゼロ行列(“-1”)が配置されるという関係が、 $j = q, q+1, q+2, \dots, q+s-1, q+s$ (ただし、 s は1以上の整数)で成立する。

【0222】

30

具体的には、図24から分かるように、22列の1行及び2行には単位行列(“0”)が配置され、22列の1行及び2行以外の行(3行及び4行)にはゼロ行列(“-1”)が配置され、かつ、23列の2行及び3行には単位行列(“0”)が配置され、23列の2行及び3行以外の行(1行及び4行)にはゼロ行列(“-1”)が配置され、かつ、24列の3行及び4行には単位行列(“0”)が配置され、24列の3行及び4行以外の行(1行及び2行)にはゼロ行列(“-1”)が配置される。そのため、図24のサブ行列 H'_{b_2} に四角枠で囲まれた部分に示すように、単位行列(“0”)が同一行に隣接して配置される。

【0223】

40

単位行列は、行列の対角要素のみが“1”であり、他の要素は“0”である。そのため、単位行列の列に対応するビットを送信しないビット(バンクチャビット)とすると、消失ビットは単位行列の各行に1ビットずつのみとなる。しかし、単位行列が、同一行に隣接して配置された場合に、単位行列を含む列に対応するビットを送信しないビット(バンクチャビット)とすると、各行において消失ビットが2ビットとなる。

【0224】

具体的には、図24における22列、23列の2行目の単位行列(“0”)のように、単位行列が、同一行に隣接して配置される場合に、これら2つの単位行列を含む列に対応するビットを送信しないビット(バンクチャビット)とすると、22列の2行目の単位行列(“0”)の各行では消失ビットが1ビットであり、23列の2行目の単位行列(“0”)の各行では消失ビットが1ビットであり、各単位行列ごとでみると消失ビットは各行

50

に 1 ビットとなるが、これら単位行列は同一行に隣接して配置されるため、単位行列が配置される同一行でみると、消失ビットは 2 ビットとなる。

【 0 2 2 5 】

上記 1) で説明したように、消失ビットは 2 ビット未満であるのが望ましい。したがって、2 ビット消失を回避するために、単位行列が同一行に隣接して配置されていない # 2 1 及び # 2 3 の列に対応するビットを送信しないビット (パンクチャビット) とするようなパンクチャパターンを用いることとする。すなわち、# 2 1 の列に対応するビットを送信しないビット (パンクチャビット) とする場合、1 サブブロック行列の列数以上離れた # 2 3 の列に対応するビットを送信しないビット (パンクチャビット) とする。このように、サブブロック行列の列数を単位として符号化系列をパンクチャする場合に、パンクチャする間隔を、1 単位 (1 サブブロック行列の列数) 以上とすることにより、単位行列又はゼロ行列から構成されるサブ行列 H'_{b_2} において、パンクチャにより消失するビットが、各行 1 ビットのみとなり、2 ビット消失を回避することができるため、受信品質の劣化を回避することができる。

【 0 2 2 6 】

一方、# 2 1 及び # 2 2 の列に対応するビット、又は、# 2 2 及び # 2 3 の列に対応するビットを送信しないビット (パンクチャビット) とするようなパンクチャパターンを用いる場合には、同一行に隣接する単位行列の列に対応するビットが送信しないビット (パンクチャビット) とされるため、2 ビット消失が起こり、受信特性が劣化してしまう。

【 0 2 2 7 】

更に、上記 2) を考慮すると、パリティ検査行列 H_b のうち、サブ行列 H'_{b_1} では、列重みが 3 以上であるため、サブ行列 H'_{b_1} の列に対応するビットを送信しないビット (パンクチャビット) とすることにより、列演算により外部値の対数尤度比の絶対値が大きくなるように更新され、消失ビットの対数尤度比が収束する可能性が高くなり、受信特性が改善される。

【 0 2 2 8 】

図 2 5 には、# 2 1, # 2 3 に加えて、# 4 の列に対応するビットが送信しないビット (パンクチャビット) とされる例が示されている。# 4 には、# 2 3 において単位行列 (“ 0 ”) が配置される行に、ゼロ行列 (“ - 1 ”) が配置されているため、# 4, # 2 1, # 2 3 の列に対応するビットを送信しないビット (パンクチャビット) とした場合に、当該行の消失ビットが 1 ビットに維持されるので、受信品質の劣化を抑えることができる。

【 0 2 2 9 】

なお、# 4, # 2 1, # 2 3 の列に対応するビットを送信しないビット (パンクチャビット) とした場合のパンクチャテーブル (パンクチャパターン) w は、式 (1 0) のようにあらわされる。

【 数 1 0 】

$$w = [\underbrace{1 \ 1 \ \dots \ 1 \ 0 \ 0 \ \dots \ 0}_{192} \underbrace{1 \ 1 \ \dots \ 1 \ 0 \ 0 \ \dots \ 0}_{48} \underbrace{1 \ 1 \ \dots \ 1 \ 0 \ 0 \ \dots \ 0}_{768} \underbrace{1 \ 1 \ \dots \ 1 \ 0 \ 0 \ \dots \ 0}_{48} \underbrace{1 \ 1 \ \dots \ 1 \ 0 \ 0 \ \dots \ 0}_{48} \underbrace{1 \ 1 \ \dots \ 1 \ 0 \ 0 \ \dots \ 0}_{48}]$$

$$\dots (10)$$

式 (1 0) において、パンクチャテーブル w に含まれる 0 は、送信しないビット (パンクチャビット) を意味する。すなわち、図 2 5 に示す例では、# 4, # 2 1, # 2 3 の列に対応するビット、すなわち、 $x_{192}, x_{193}, \dots, x_{238}, x_{239}, p_{48}, p_{49}, \dots, p_{94}, p_{95}, p_{144}, p_{145}, \dots, p_{190}, p_{191}$ がパンクチャされる。

【 0 2 3 0 】

以上のように、本実施の形態では、パリティ検査行列 H_b を、サブブロック行列を 1 単位として表現したとき、 j 列の i 行及び $(i + 1)$ 行には単位行列 (“ 0 ”) が配置され

、 j 列の i 行及び $(i+1)$ 行以外の行にはゼロ行列（“ - 1 ”）が配置され、かつ、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行には単位行列（“ 0 ”）が配置され、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行以外の行にはゼロ行列（“ - 1 ”）が配置されるという関係が、 $j = q, q+1, q+2, \dots, q+s-1, q+s$ （ただし、 s は 1 以上の整数）で成立するサブ行列 H'_{b_2} に対し、サブブロック行列の列数を単位として送信しないビット（パンクチャビット）を選択する場合に、1 単位（サブブロック行列の列数）以上間隔をあけて送信しないビット（パンクチャビット）とするようにした。

【0231】

更に、パリティ検査行列 H_b のうち、上記サブ行列 H'_{b_2} 以外のサブ行列 H'_{b_1} において、列重みが 3 以上の列に対応するビットを送信しないビット（パンクチャビット）とすることにより、列演算における外部値の対数尤度比の絶対値が大きくなるよう更新が行われるようになるため、受信品質の劣化を抑えることができるようになる。

【0232】

なお、列重みが 3 以上の列に対応するビットを送信しないビット（パンクチャビット）とする場合、サブ行列 H'_{b_2} において、#21 及び #23 の列に対応するビットが送信しないビット（パンクチャビット）とされる場合には、例えば、#4 のように、#21 又は #23 に含まれる単位行列のいずれかの行に配置されるゼロ行列の列に対応するビットを送信しないビット（パンクチャビット）とする。このようにして、#4, #21, #23 の列に対応するビットを送信しないビット（パンクチャビット）とした場合、#4 においてゼロ行列が配置される行の消失ビットを 1 ビットに抑えることができるため、受信品質の劣化を抑えることができる。

【0233】

以上、符号化率 5/6 の式 (9) の検査行列をもつ QC-LDPC 符号から、パンクチャにより、符号化率 20/21 を実現するための好適なパンクチャパターンについて説明した。また、別の例として、符号化率 5/6 の式 (9) の検査行列をもち、サブブロック行列 216×216 、つまり、Information size= 4320 ビットの QC-LDPC 符号から、パンクチャにより、符号化率 20/21 を実現するための好適なパンクチャテーブル（パンクチャパターン） w は、例えば、式 (11-1) ~ 式 (11-3) のようにあらわされる。

【数 11】

$$w = \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \underbrace{[1 \ 1 \ \dots \ 1]_{4320}}_{4320} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \underbrace{[1 \ 1 \ \dots \ 1]_{216}}_{216} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \dots (11-1)$$

$$w = \underbrace{[1 \ 1 \ \dots \ 1]_{432}}_{432} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \underbrace{[1 \ 1 \ \dots \ 1]_{3888}}_{3888} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \underbrace{[1 \ 1 \ \dots \ 1]_{216}}_{216} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \dots (11-2)$$

$$w = \underbrace{[1 \ 1 \ \dots \ 1]_{864}}_{864} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \underbrace{[1 \ 1 \ \dots \ 1]_{3456}}_{3456} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \underbrace{[1 \ 1 \ \dots \ 1]_{216}}_{216} \underbrace{[0 \ 0 \ \dots \ 0]_{216}}_{216} \dots (11-3)$$

【0234】

なお、以上の説明では、サブブロック行列の列数を単位として符号化系列をパンクチャする場合について説明したが、1 単位（サブブロック行列の列数）以上間隔を開けて、サブブロック行列の列数を単位として送信しないビット（パンクチャビット）の候補を設定し、この候補から、送信しないビット（パンクチャビット）を決定するようにしても良い

。この場合において、送信しないビット（パンクチャビット）の候補は、サブブロック行列の列数を単位として設定し、かつ、上述したように、同一行に隣接する単位行列を含まない列に対応するビットとする。

【 0 2 3 5 】

例えば、図 2 6 に示すように、# 4 , # 2 1 及び # 2 3 の列に対応するビットを送信しないビット（パンクチャビット）の候補とし、# 4 , # 2 1 及び # 2 3 の列に対応する x_{192} , x_{193} , ... , x_{238} , x_{239} , p_{48} , p_{49} , ... , p_{94} , p_{95} , p_{144} , p_{145} , ... , p_{190} , p_{191} のうち、点線の丸で囲まれた一部のビットを送信しないビット（パンクチャビット）に決定するようにしてもよい。

【 0 2 3 6 】

この方法を用いて、符号化率 5 / 6 の式 (9) の検査行列をもち、サブブロック行列 48×48 、つまり、Information size= 960ビットの QC-LDPC 符号から、パンクチャにより、符号化率 1 6 / 1 8 を実現するための好適なパンクチャテーブル（パンクチャパターン） w は、式 (1 2) のようにあらわされる。

【 数 1 2 】

$$w = [\underbrace{1 \ 1 \ \dots \ 1}_{720} \ \underbrace{0 \ 0 \ \dots \ 0}_{24} \ \underbrace{1 \ 1 \ \dots \ 1}_{288} \ \underbrace{0 \ 0 \ \dots \ 0}_{24} \ \underbrace{1 \ 1 \ \dots \ 1}_{72} \ \underbrace{0 \ 0 \ \dots \ 0}_{24}]$$

... (1 2)

【 0 2 3 7 】

また、別の例として、符号化率 5 / 6 の式 (9) の検査行列をもち、サブブロック行列 216×216 、つまり、Information size= 4320ビットの QC-LDPC 符号から、パンクチャにより、符号化率 1 6 / 1 8 を実現するための好適なパンクチャテーブル（パンクチャパターン） w は、式 (1 3) のようにあらわされる。

【 数 1 3 】

$$w = [\underbrace{1 \ 1 \ \dots \ 1}_{2808} \ \underbrace{0 \ 0 \ \dots \ 0}_{108} \ \underbrace{1 \ 1 \ \dots \ 1}_{1620} \ \underbrace{0 \ 0 \ \dots \ 0}_{108} \ \underbrace{1 \ 1 \ \dots \ 1}_{432} \ \underbrace{0 \ 0 \ \dots \ 0}_{108}]$$

... (1 3)

【 0 2 3 8 】

これらの場合においても、各行に消失ビットが 2 ビット発生するのを回避することができるので、良好な受信品質を得ることができるとともに、パンクチャ後の符号化率を柔軟に設定することができる。

【 0 2 3 9 】

また、以上の説明では、LDPC 符号の codeword v を、 $v = [x_0, x_1, \dots, x_{958}, x_{959}, p_0, p_1, \dots, p_{190}, p_{191}]$ とあらわし説明したが、情報系列及びパリティ系列の並び順は、これに限られず（例えば、 $v = [p_0, p_1, \dots, p_{190}, p_{191}, x_0, x_1, \dots, x_{958}, x_{959}]$ であってもよく、情報とパリティの並び順は一意には決定しない。） $H_p v = 0$ の対応関係、換言すると、パリティ検査行列 H_p に対応するパンクチャパターンと codeword v との対応関係から、送信しないビット（パリティビット）を決定すれば良い。

【 0 2 4 0 】

例えば、# 4 , # 2 1 , # 2 3 の列に対応するビットを送信しないビット（パンクチャビット）とする場合に、codeword v が、 $v = [p_{144}, p_{145}, \dots, p_{190}, p_{191}, x_0, x_1, \dots, x_{958}, x_{959}, p_0, p_1, \dots, p_{46}, p_{47}]$ とあらわされる場合には、 x_{144} , x_{145} , ... , x_{190} , x_{191} のうち、点線の丸で囲まれた一部のビットを送信しないビット（パンクチャビット）に決定するようにしてもよい。

4 5 , ... , x 1 9 0 , x 1 9 1 , p 0 , p 1 , ... , p 4 6 , p 4 7 , p 9 6 , p 9 7 , ... , p 1 4 2 , p 1 4 3 を送信しないビット（パンクチャビット）とすれば良い。

【 0 2 4 1 】

上述の例では、サブ行列 H'_{b_2} には、 j 列の i 行及び $(i+1)$ 行には単位行列（“0”）が配置され、 j 列の i 行及び $(i+1)$ 行以外の行にはゼロ行列（“-1”）が配置され、かつ、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行には単位行列（“0”）が配置され、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行以外の行にはゼロ行列（“-1”）が配置されるという関係が、 $j = q, q+1, q+2, \dots, q+s-1, q+s$ （ただし、 s は 1 以上の整数）で成立し、単位行列が同一行に隣接して配置される場合について説明したが、単位行列に代えて、単位行列の巡回シフト行列が同一行に隣接して配置される場合に対しても、上述のパンクチャパターンの生成方法を用いて、好適なパンクチャパターンを得ることができる。ただし、サブ行列 H'_{b_2} に、単位行列の巡回シフト行列が同一行に隣接して配置される場合には、符号化器の構成が複雑になる可能性がある。

10

【 0 2 4 2 】

なお、式（8）の符号化率 $1/2$ の QC-LDPC 符号に対しても、同様の方法により、好適なパンクチャパターンを設定することができる。図 27 を用いて、符号化率 $1/2$ の場合について説明する。

【 0 2 4 3 】

符号化率 $5/6$ の場合と同様に、式（8）の符号化率 $1/2$ の QC-LDPC 符号のパリティ検査行列 H_b に対し、サブ行列 H'_{b_1} には、サブ行列 H'_{b_1} の列数をパンクチャ周期とするパンクチャパターン # p 1 を用い、サブ行列 H'_{b_2} には、サブ行列 H'_{b_2} の列数をパンクチャパターン周期とするパンクチャパターン # p 2 を用いる。

20

【 0 2 4 4 】

図 27 は、式（8）の符号化率 $1/2$ の QC-LDPC 符号のパリティ検査行列 H_b である。式（8）の検査行列 H_b は、12 行 24 列のサブブロック行列から構成される。図 27 において、サブ行列 H'_{b_2} は、単位行列及びゼロ行列により構成され、サブ行列 H'_{b_1} は、サブ行列 H'_{b_2} 以外である。

【 0 2 4 5 】

また、図 27 において、# 0 ~ # 23 は、各列に対応する部分行列を示し、サブ行列 H'_{b_1} は、# 0 ~ # 12 により構成され、サブ行列 H'_{b_2} は、# 13 ~ # 23 により構成される。# 13 ~ # 23 は、単位行列（“0”）及びゼロ行列（“-1”）により構成される。このように、式（8）に示す QC-LDPC 符号の検査行列 H_b は、単位行列及びゼロ行列で構成されるサブ行列 H'_{b_2} を含んでいる。

30

【 0 2 4 6 】

以下では、サブブロック行列の列数を単位として符号化系列をパンクチャする場合を例に説明する。

【 0 2 4 7 】

式（8）のパリティ検査行列 H_b を、サブブロック行列を 1 単位として表現したとき、サブ行列 H'_{b_2} では、 j 列の i 行及び $(i+1)$ 行には単位行列（“0”）が配置され、 j 列の i 行及び $(i+1)$ 行以外の行にはゼロ行列（“-1”）が配置され、かつ、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行には単位行列（“0”）が配置され、 $(j+1)$ 列の $(i+1)$ 行及び $(i+2)$ 行以外の行にはゼロ行列（“-1”）が配置されるという関係が、 $j = q, q+1, q+2, \dots, q+s-1, q+s$ （ただし、 s は 1 以上の整数）で成立する。

40

【 0 2 4 8 】

具体的には、図 27 から分かるように、14 列の 1 行及び 2 行には単位行列（“0”）が配置され、14 列の 1 行及び 2 行以外の行（3 ~ 12 行）にはゼロ行列（“-1”）が配置され、かつ、15 列の 2 行及び 3 行には単位行列（“0”）が配置され、15 列の 2 行及び 3 行以外の行（1 行、4 ~ 12 行）にはゼロ行列（“-1”）が配置され、かつ、

50

・・・、24列の11行及び12行には単位行列(“0”)が配置され、24列の11行及び12行以外の行(1～10行)にはゼロ行列(“-1”)が配置される。そのため、図27のサブ行列 H'_{b_2} に四角枠で囲まれた部分に示すように、単位行列(“0”)が同一行に隣接して配置される。

【0249】

符号化率1/2の場合も符号化率5/6の場合と同様に、同一行に単位行列が隣接して配置されていない列に対応するビットを送信しないビット(パンクチャビット)とするようパンクチャパターンを設定する。例えば、サブブロック行列の列数を単位として符号化系列をパンクチャし、#20の列に対応するビットを送信しないビット(パンクチャビット)とする場合、1サブブロック行列の列数以上離れた#15及び#23の列に対応するビットを送信しないビット(パンクチャビット)とする。このように、サブブロック行列の列数を単位として符号化系列をパンクチャする場合に、パンクチャする間隔を、1単位(1サブブロック行列の列数)以上とする。これにより、単位行列又はゼロ行列から構成されるサブ行列 H'_{b_2} において、パンクチャにより消失するビットが、各行1ビットのみとなり、2ビット消失を回避することができ、受信品質の劣化を回避することができる。

10

【0250】

また、例えば、#15、#20の列に対応するビットを送信しないビット(パンクチャビット)としても良い。#15と#20とは、1単位以上離れている。また、#20、#23の列に対応するビットを送信しないビット(パンクチャビット)としても良い。#20と#23とは、1単位以上離れている。

20

【0251】

なお、#15、#20、#23の列に対応するビットの全てを送信しないビット(パンクチャビット)としなくても良く、#15、#20、#23の列に対応するビットを送信しないビット(パンクチャビット)の候補とし、符号化率に応じて、この候補から送信しないビット(パンクチャビット)を決定するようにしても良い。このようにすることで、各行の消失ビットを1ビットに抑え良好な受信品質を得ることができるとともに、パンクチャ後の符号化率を柔軟に設定することができる。

【0252】

また、更に、パリティ検査行列 H_b のうち、サブ行列 H'_{b_1} では、列重みが3以上であるため、サブ行列 H'_{b_1} の列に対応するビットを送信しないビット(パンクチャビット)とすることにより、列演算により外部値の大きさが適切に更新され、消失ビットの対数尤度比が適切に得られるようになり、受信特性が改善される。

30

【0253】

図27には、サブ行列 H'_{b_1} において、#10の列に対応するビットが送信しないビット(パンクチャビット)とされる例が示されている。#10には、#15、#20、#23において単位行列(“0”)が配置される行に、ゼロ行列(“-1”)が配置されているため、#10、#15、#20、#23の列に対応するビットを送信しないビット(パンクチャビット)とした場合に、当該行の消失ビットが1ビットに維持され、受信品質の劣化を抑えることができると可能性が高い。

40

【0254】

なお、以上の説明では、サブブロック行列の列数を単位として、サブ行列 H'_{b_1} から1サブブロック行列を選択し、サブ行列 H'_{b_2} から複数のサブブロック行列を選択し、選択したサブブロック行列の列に対応するビットを送信しないビット(パンクチャビット)又は送信しないビット(パンクチャビット)の候補とする場合について説明したが、各サブ行列からの選択数は、これに限られず、サブ行列 H'_{b_1} から複数のサブブロック行列を選択するようにしても良い。

【0255】

本発明は、QC-LDPC符号のように、検査行列、生成行列に規則性があるような場合に有効である。

50

【 0 2 5 6 】

本発明は上記全ての実施の形態に限定されず、種々変更して実施することが可能である。例えば、上記実施の形態では、主に、符号化器で実現する場合について説明しているが、これに限られるものではなく、電灯線通信装置で実現する場合においても適用可能である。

【 0 2 5 7 】

また、この符号化方法は、ソフトウェアとして行うことも可能である。例えば、上記符号化方法を実行するプログラムは、予め R O M (Read Only Memory) に格納しておき、そのプログラムを C P U (Central Processor Unit) によって動作させるようにしても良い。

10

【 0 2 5 8 】

また、上記符号化方法を実行するプログラムをコンピュータで読み取り可能な記憶媒体に格納し、記憶媒体に格納されたプログラムをコンピュータの R A M (Random Access Memory) に記録して、コンピュータをそのプログラムにしたがって動作させるようにしても良い。

【 0 2 5 9 】

また、本発明は、無線通信に限らず、電灯線通信 (P L C : Power Line Communication)、可視光通信、光通信においても有用であることは言うまでもない。

【産業上の利用可能性】

【 0 2 6 0 】

20

本発明は、例えば、Q C - L D P C 符号等のブロック符号を用いる場合に、受信品質を向上させつつ、伝送量を低減させ、伝送効率の劣化を抑圧することができ、Q C - L D P C 等のように、ゼロ行列を部分的に含むパリティ生成行列を用いて符号化系列を形成する符号化器、送信装置及び符号化方法として有用である。

【図面の簡単な説明】

【 0 2 6 1 】

【図 1】ブロック符号のブロック構成例及び実際に送信する符号化系列を示す図

【図 2】本発明の実施の形態 1 にかかる通信装置に用いられる符号化器の入出力データを示す図

【図 3】Q C - L D P C 符号のパリティ生成行列 g の一例を示す図

30

【図 4】ゼロ行列の一例を示す図

【図 5】ゼロ行列の別の一例を示す図

【図 6】実施の形態 1 に係る符号化器の構成例を示す図

【図 7】実施の形態 1 に係る復号化器の構成例を示す図

【図 8】通信装置 # 1 が送信する変調信号のフレーム構成例を示す図

【図 9】実施の形態 1 に係る符号化器を有する通信装置 # 1 の構成例を示す図

【図 10】実施の形態 1 に係る復号化器を有する通信装置 # 2 の構成例を示す図

【図 11】Q C - L D P C 符号の 1 ブロックの構成例を示す図

【図 12】本発明の実施の形態 2 に係る符号化器の構成例を示す図

【図 13】情報ビットの配置例を示す図

40

【図 14】データ長 と送信するパリティビットの削減方法との対応を示す図

【図 15】本発明の実施の形態 3 に係る符号化器の構成例を示す図

【図 16 A】パンクチャパターンの切り替え方法を説明するための図

【図 16 B】パンクチャパターンの切り替え方法を説明するための別の図

【図 16 C】パンクチャパターンの切り替え方法を説明するための別の図

【図 17】制御情報の配置例を示す図

【図 18】本発明の実施の形態 4 における制御情報の配置例を示す図

【図 19 A】本発明の実施の形態 5 におけるパンクチャパターンの適用例を示す図

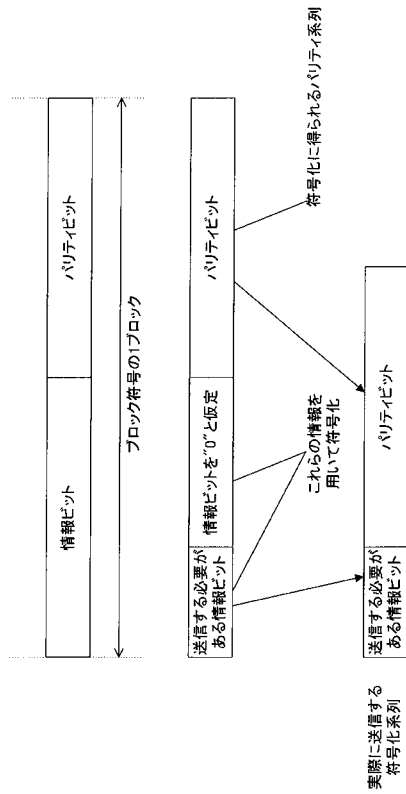
【図 19 B】実施の形態 5 におけるパンクチャパターンの別の適用例を示す図

【図 19 C】実施の形態 5 におけるパンクチャパターンの別の適用例を示す図

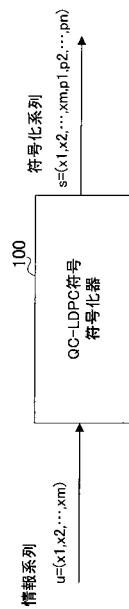
50

【図 2 0 A】実施の形態 5 におけるパンクチャパターンの別の適用例を示す図	
【図 2 0 B】実施の形態 5 におけるパンクチャパターンの別の適用例を示す図	
【図 2 0 C】実施の形態 5 におけるパンクチャパターンの別の適用例を示す図	
【図 2 1】本発明の実施の形態 6 におけるパンクチャパターンの適用例を示す図	
【図 2 2】実施の形態 6 におけるパンクチャパターンの別の適用例を示す図	
【図 2 3】本発明の実施の形態 7 におけるパンクチャパターンを説明するための図	
【図 2 4】符号化率 5 / 6 の Q C - L D P C 符号の検査行列 H_b を示す図	
【図 2 5】実施の形態 7 におけるパンクチャパターンの一例を示す図	
【図 2 6】実施の形態 7 におけるパンクチャパターンの別の一例を示す図	
【図 2 7】符号化率 1 / 2 の Q C - L D P C 符号の検査行列 H_b 及びパンクチャパターン の一例を示す図	10
【符号の説明】	
【0 2 6 2】	
1 0 0 , 1 0 0 a , 6 0 0 符号化器	
1 1 0 , 1 1 0 a ゼロ行列設定部	
1 2 0 , 1 2 0 a 配置部	
1 3 0 , 6 1 0 符号化部	
1 4 0 , 6 3 0 パンクチャ部 (データ削減部)	
3 0 0 復号化器	
3 1 0 固定対数尤度比挿入部	20
3 2 0 B P 復号部	
4 0 0 , 5 0 0 通信装置	
4 1 0 符号化部	
4 2 0 インタリーバ	
4 3 0 マッピング部	
4 4 0 送信部	
5 1 0 受信部	
5 2 0 制御情報検出部	
5 3 0 対数尤度比算出部	
5 4 0 デインタリーバ	30
5 5 0 復号化部	
6 2 0 パンクチャパターン設定部	

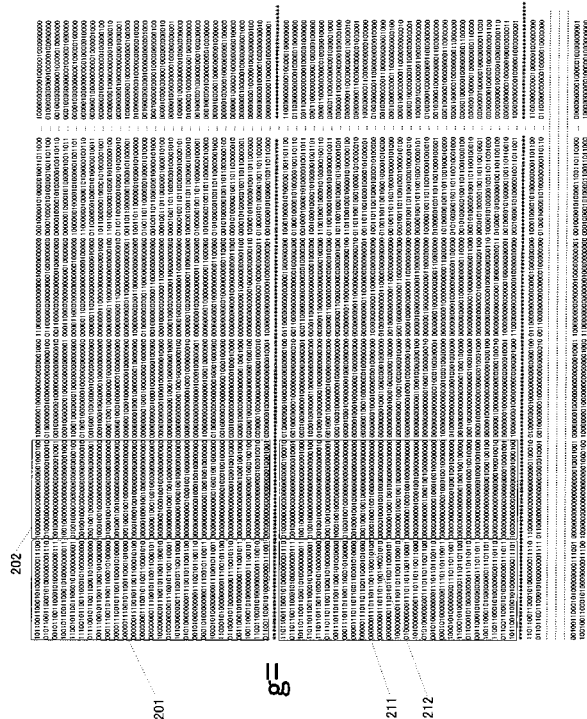
【図 1】



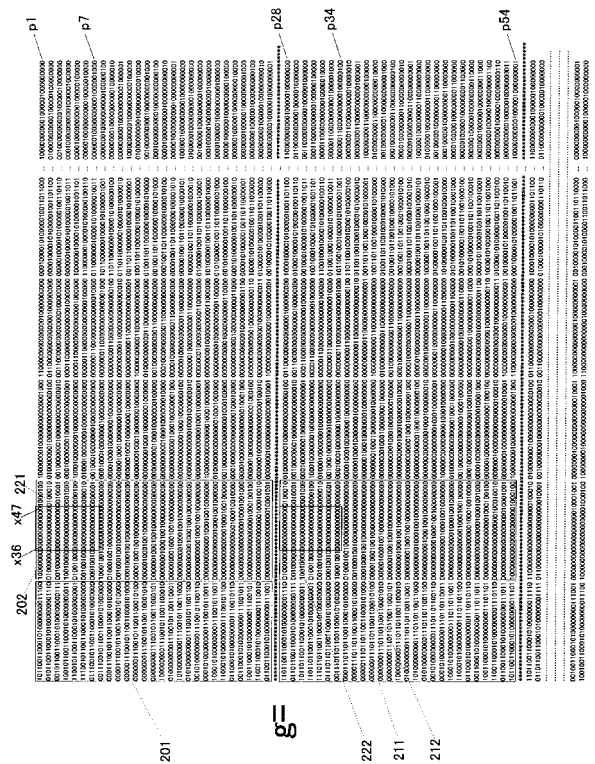
【図 2】



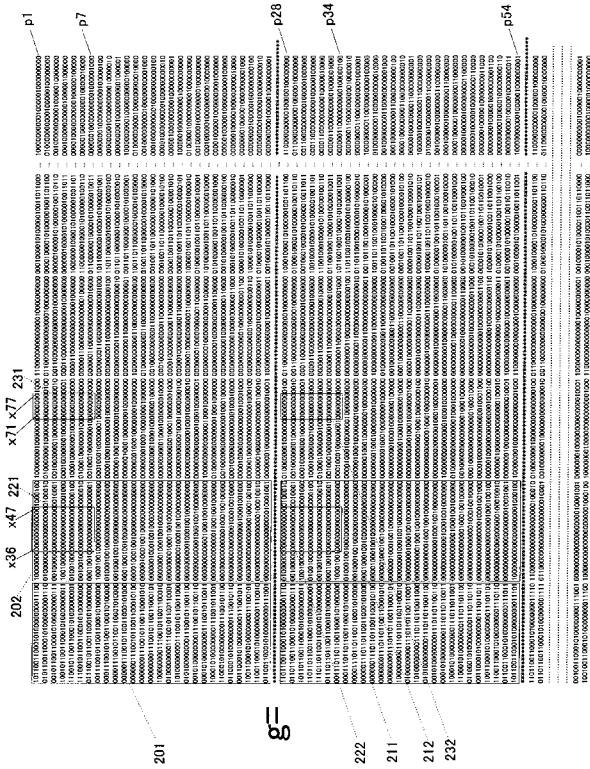
【図 3】



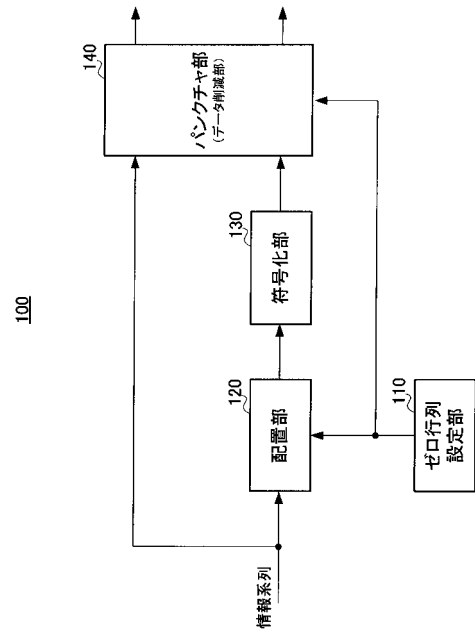
【図 4】



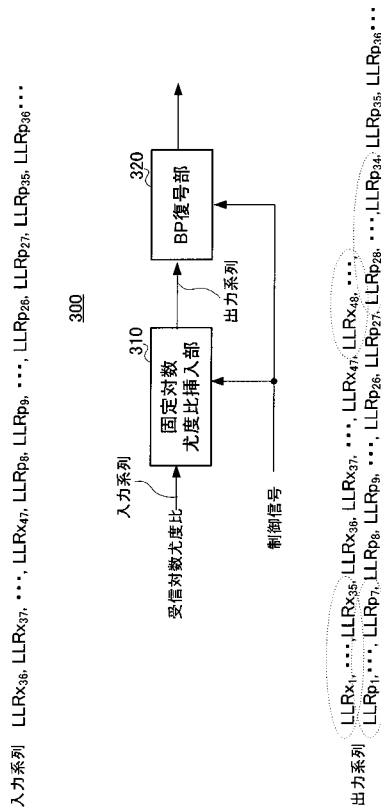
【図 5】



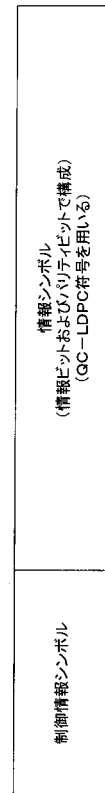
【図 6】



【図 7】

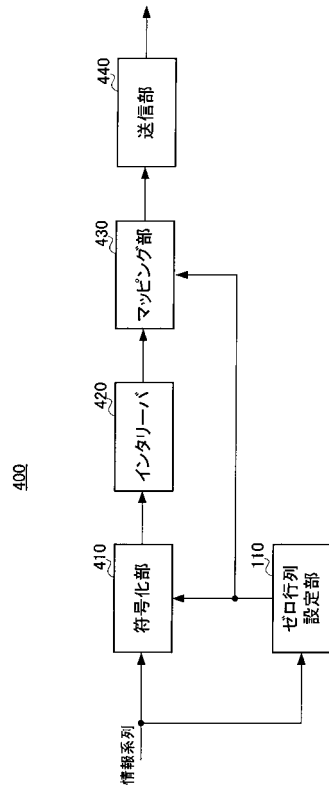


【図 8】

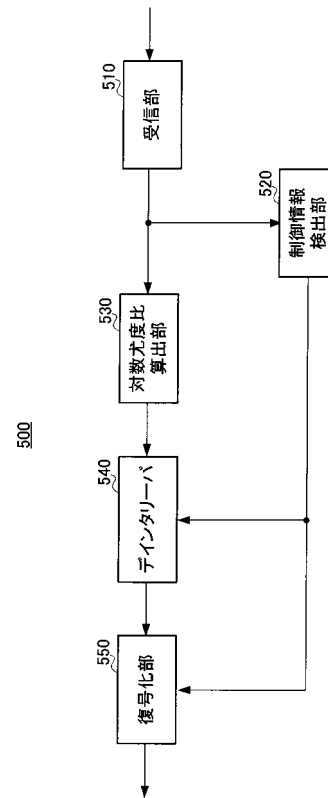


時間 →

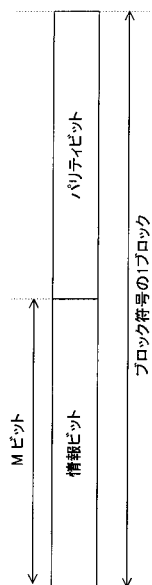
【図 9】



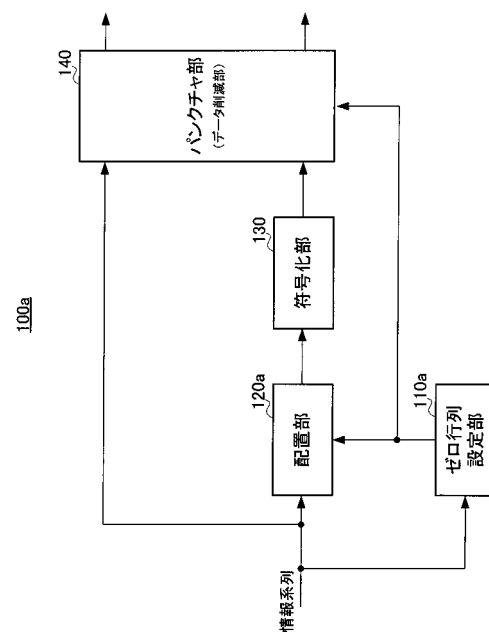
【図 10】



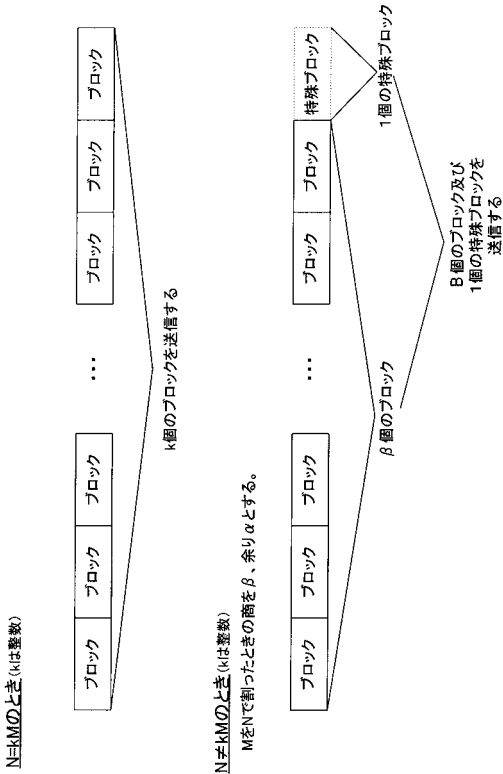
【図 11】



【図 12】



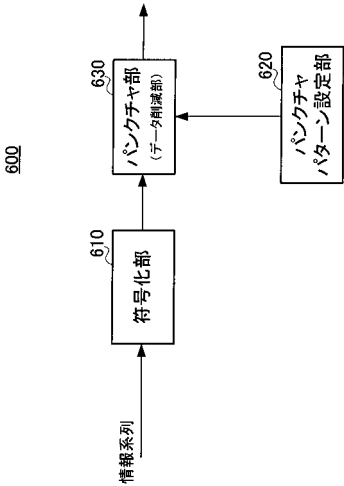
【図 1 3】



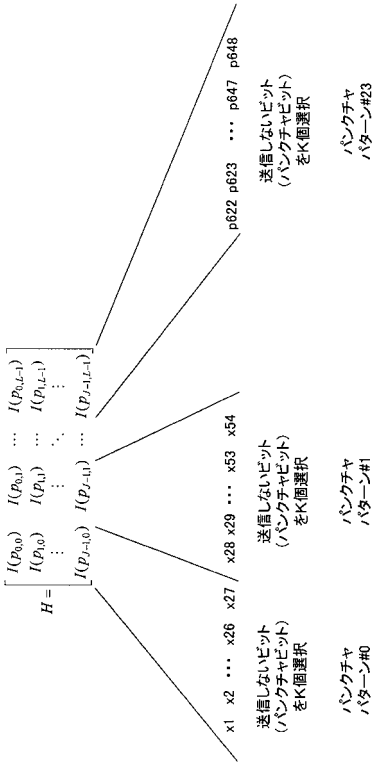
【図 1 4】

$0 < \alpha \leq a1$	$a1 < \alpha \leq a2$	$a2 < \alpha \leq M-1$
ゼロ行列#1 (送信するパリティビット削減方法#1)	ゼロ行列#2 (送信するパリティビット削減方法#2)	ゼロ行列無し (送信するパリティビットの削減は行わない)

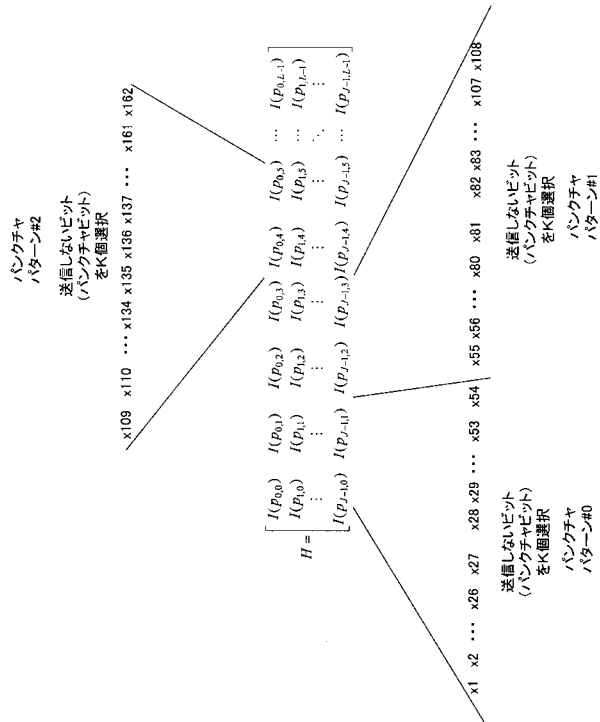
【図 1 5】



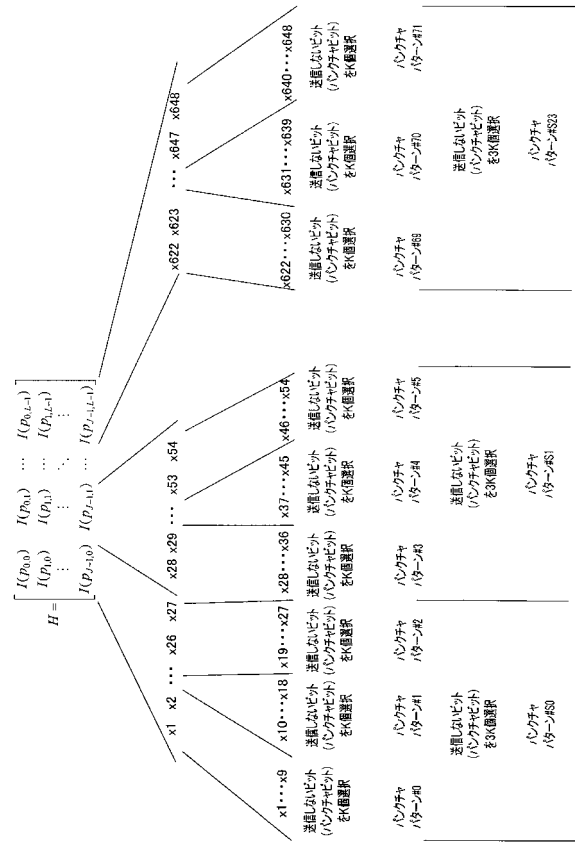
【図 1 6 A】



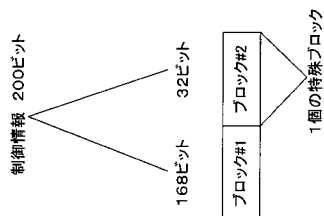
【図 16B】



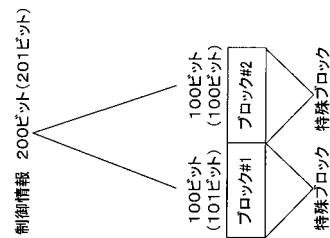
【図 16C】



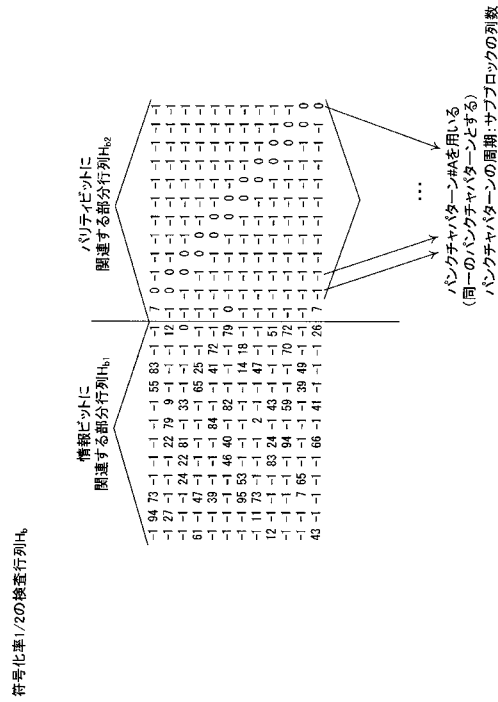
【図 17】



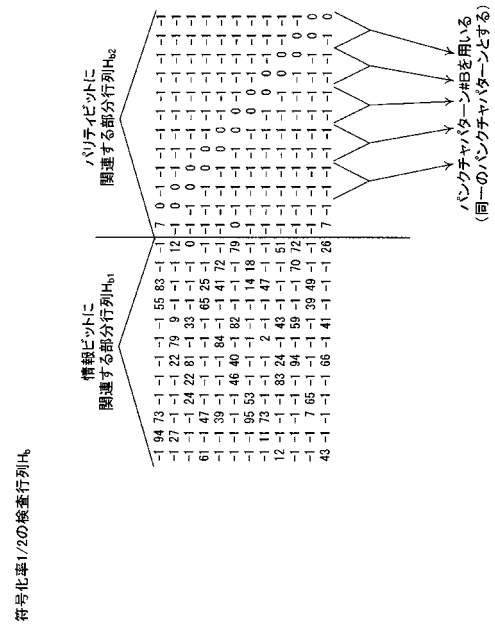
【図 18】



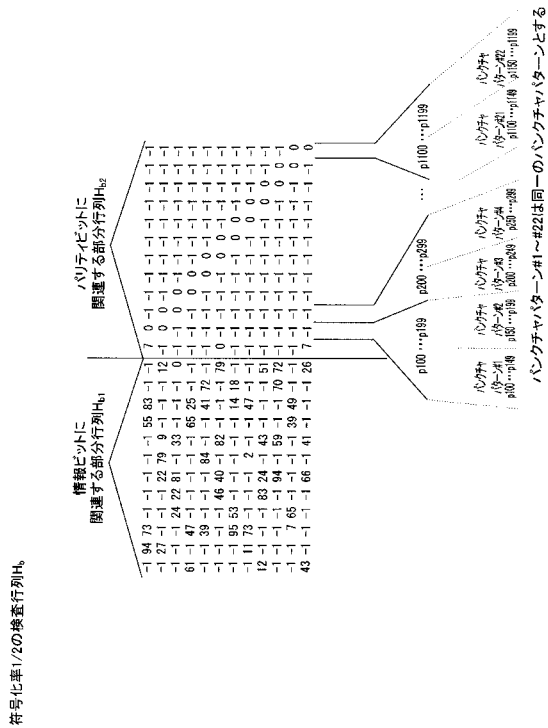
【図 19 A】



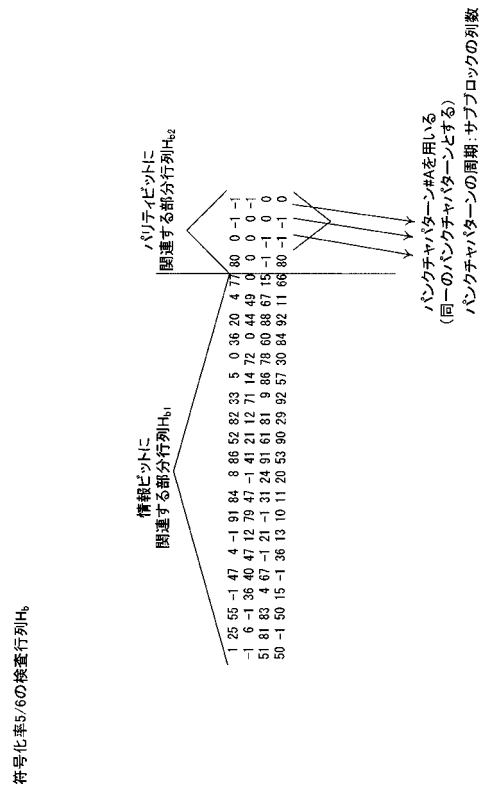
【図 19 B】

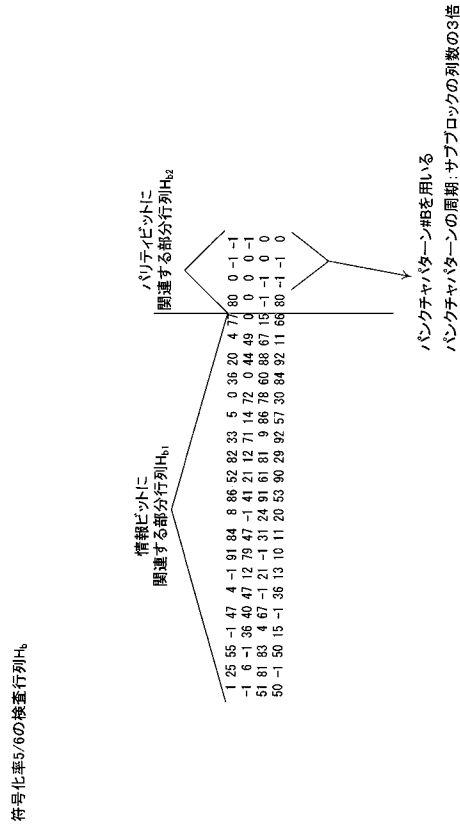


【図 19 C】



【図 20 A】

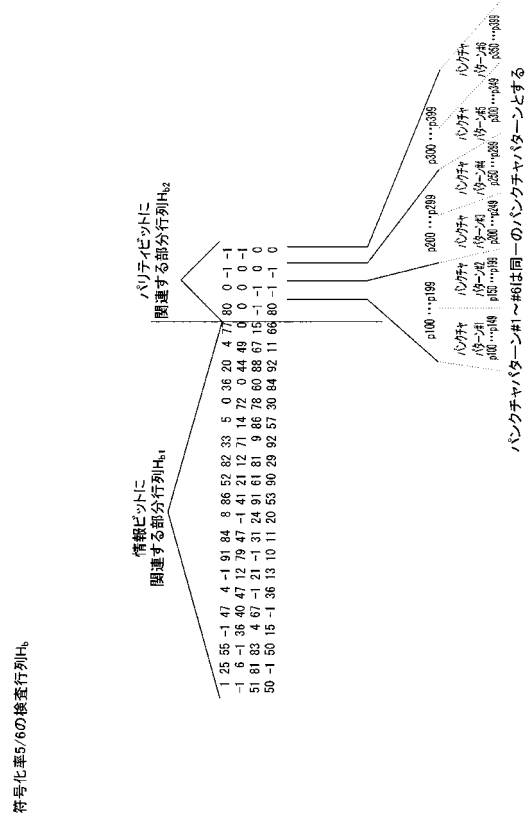




$w = [101111101 \ 111111110 \ 0111101100 \ 1011111101 \ 0011111111]$

$vi = [\ s63*1, \ s50*1, \ s50*2, \ s50*3, \ s50*4, \ s50*5, \ s50*6, \ s50*7, \ s50*8, \ s50*9, \ s50*10, \ s50*11, \ s50*12, \ s50*13, \ s50*14, \ s50*15, \ s50*16, \ s50*17, \ s50*18, \ s50*19, \ s50*20, \ s50*21, \ s50*22, \ s50*23, \ s50*24, \ s50*25, \ s50*26, \ s50*27, \ s50*28, \ s50*29, \ s50*30, \ s50*31, \ s50*32, \ s50*33, \ s50*34, \ s50*35, \ s50*36, \ s50*37, \ s50*38, \ s50*39, \ s50*40, \ s50*41, \ s50*42, \ s50*43, \ s50*44, \ s50*45, \ s50*46, \ s50*47, \ s50*48, \ s50*49]$

削除されるビット



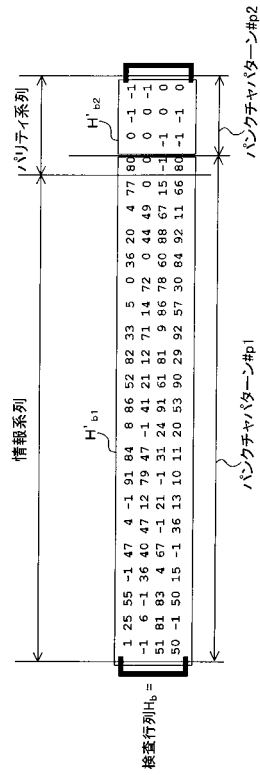
$w = [1110111111 \ 1111111101 \ 0111111011 \ 1111111110 \ 1111111111 \ 0111111111 \ 0111]$

$vi = [\ s63*1, \ s63*2, \ s63*3, \ s63*4, \ s63*5, \ s63*6, \ s63*7, \ s63*8, \ s63*9, \ s63*10, \ s63*11, \ s63*12, \ s63*13, \ s63*14, \ s63*15, \ s63*16, \ s63*17, \ s63*18, \ s63*19, \ s63*20, \ s63*21, \ s63*22, \ s63*23, \ s63*24, \ s63*25, \ s63*26, \ s63*27, \ s63*28, \ s63*29, \ s63*30, \ s63*31, \ s63*32, \ s63*33, \ s63*34, \ s63*35, \ s63*36, \ s63*37, \ s63*38, \ s63*39, \ s63*40, \ s63*41, \ s63*42, \ s63*43, \ s63*44, \ s63*45, \ s63*46, \ s63*47, \ s63*48, \ s63*49, \ s63*50, \ s63*51, \ s63*52, \ s63*53, \ s63*54, \ s63*55, \ s63*56, \ s63*57, \ s63*58, \ s63*59, \ s63*60, \ s63*61, \ s63*62]$

削除されるビット

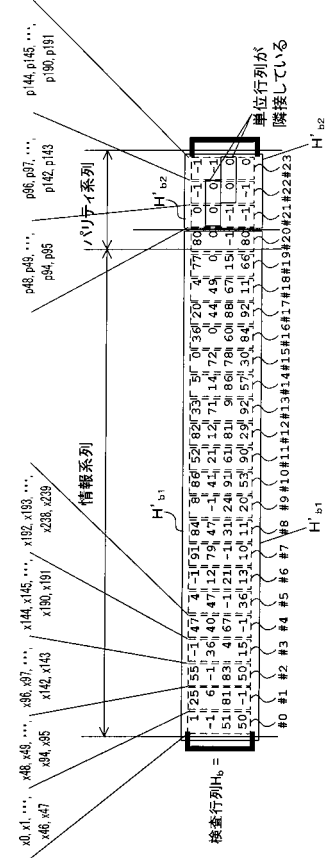
【 図 2 3 】

符号化率 $R=5/6$ の検査行列 H_b



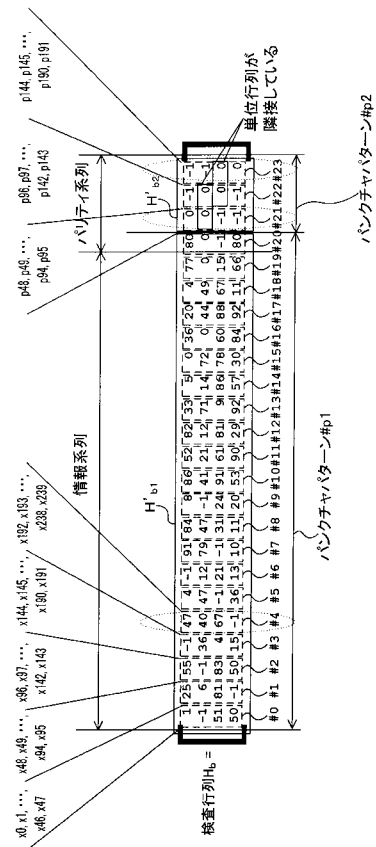
【 図 2 4 】

符号化率 $R=5/6$ の検査行列 H_b



【 図 2 5 】

符号化率 $R=5/6$ の検査行列 H_b



【 図 2 6 】

符号化率 $R=5/6$ の検査行列 H_b 